N 6

```
NNN         NNN     CCCCCCCCCCCC  PPPPPPPPPPPP
NNN         NNN     CCCCCCCCCCCC  PPPPPPPPPPPP
NNN         NNN     CCCCCCCCCCCC  PPPPPPPPPPPP
NNN         NNN  CCC              PPP         PPP
NNN         NNN  CCC              PPP         PPP
NNN         NNN  CCC              PPP         PPP
NNNNNN      NNN  CCC              PPP         PPP
NNNNNN      NNN  CCC              PPP         PPP
NNNNNN      NNN  CCC              PPP         PPP
NNN   NNN   NNN  CCC              PPPPPPPPPPPP
NNN   NNN   NNN  CCC              PPPPPPPPPPPP
NNN   NNN   NNN  CCC              PPPPPPPPPPPP
NNN      NNNNNN  CCC              PPP
NNN      NNNNNN  CCC              PPP
NNN      NNNNNN  CCC              PPP
NNN         NNN  CCC              PPP
NNN         NNN  CCC              PPP
NNN         NNN  CCC              PPP
NNN         NNN     CCCCCCCCCCCC  PPP
NNN         NNN     CCCCCCCCCCCC  PPP
NNN         NNN     CCCCCCCCCCCC  PPP
```

FILEID**NCPNETIO

```
NN      NN    CCCCCCCC  PPPPPPPP    NN      NN  EEEEEEEEEE  TTTTTTTTTT    IIIIII      OOOOOO
NN      NN    CCCCCCCC  PPPPPPPP    NN      NN  EEEEEEEEEE  TTTTTTTTTT    IIIIII      OOOOOO
NN      NN  CC          PP      PP  NN      NN  EE              TT          II      OO      OO
NN      NN  CC          PP      PP  NN      NN  EE              TT          II      OO      OO
NNNN    NN  CC          PP      PP  NNNN    NN  EE              TT          II      OO      OO
NNNN    NN  CC          PP      PP  NNNN    NN  EE              TT          II      OO      OO
NN  NN  NN  CC          PPPPPPPP    NN  NN  NN  EEEEEEEE        TT          II      OO      OO
NN  NN  NN  CC          PPPPPPPP    NN  NN  NN  EEEEEEEE        TT          II      OO      OO
NN    NNNN  CC          PP          NN    NNNN  EE              TT          II      OO      OO
NN    NNNN  CC          PP          NN    NNNN  EE              TT          II      OO      OO
NN      NN  CC          PP          NN      NN  EE              TT          II      OO      OO
NN      NN  CC          PP          NN      NN  EE              TT          II      OO      OO      ....
NN      NN    CCCCCCCC  PP          NN      NN  EEEEEEEEEE      TT        IIIIII      OOOOOO        ....
NN      NN    CCCCCCCC  PP          NN      NN  EEEEEEEEEE      TT        IIIIII      OOOOOO        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 %TITLE  'Network I/O Routines'
0 MODULE NCPNETIO (IDENT = 'V04-000',
0                 ADDRESSING_MODE(EXTERNAL=GENERAL),
0                 ADDRESSING_MODE(NONEXTERNAL=GENERAL)) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     Network Control Program (NCP)
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the routines which establish and communicate
1 !       over a logical link to NML on the executor node.
1 !
1 ! ENVIRONMENT:  VAX/VMS Operating System
1 !
1 ! AUTHOR:       Darrell Duffy   , CREATION DATE: 30-October-1979
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-005 PRD0097         Paul R. DeStefano       19-Apr-1984
1 !               Disable defaulting of area to 1.n if area is not
1 !               specified (defeats PRD0087).
1 !
1 !       V03-004 PRD0094         Paul R. DeStefano       09-Apr-1984
1 !               Modified NCP$READRSP and NCP$CONERR routines to
1 !               interpret the error detail when the error status
1 !               is NMA$C_STS_OPE (operation failure).  Additional
1 !               text is now appended to the original error text
1 !               in the nice message based on the detail.
1 !
1 !       V03-003 PRD0087         Paul R. DeStefano       27-Mar-1984
```

```
1 !                     Make SET EXEC NODE n default to 1.n if area is not
1 !                     specified.
1 !
1 !             V03-002 RPG0002         Bob Grosso              20-Apr-1983
1 !                     Add NCP$CONERR for the CONNECT routine to use to
1 !                     process errors.
1 !
1 !             V03-001 RPG0001         Bob Grosso              16-Mar-1983
1 !                     Update version number checking for version IV.
1 !
1 !             V009    TMH0009         Tim Halvorsen           11-Jan-1982
1 !                     Save verison number of NML server in LCB.
1 !                     Make NCP$OPENLINK a global routine.
1 !
1 !             V008    TMH0008         Tim Halvorsen           15-Dec-1981
1 !                     Print detail messages for FCO NICE errors.
1 !
1 !             V007    TMH0007         Tim Halvorsen           22-Oct-1981
1 !                     Fix the spelling on some messages.
1 !
1 !             V006    LMK0006         Len Kawell              19-Sep-1981
1 !                     Change version checking to allow current or greater and V2.0.
1 !
1 !             V005    TMH0005         Tim Halvorsen           11-Aug-1981
1 !                     Use different detail text table if looking up system-
1 !                     specific entity number.  When formatting a parameter
1 !                     detail, use the signed entity in NCP$GL_ENTITY rather
1 !                     than the option byte, since it doesn't tell whether
1 !                     its a system-specific entity or not.  Only supply a
1 !                     comma following an NML response message if there is
1 !                     a non-blank detail following it.
1 !
1 !             V004    TMH0004         Tim Halvorsen           10-Jul-1981
1 !                     Change all non-local references to use general addressing.
1 !                     Use new callable NML whenever we are communicating with the
1 !                     local node without any access control string.
1 !
1 !             V003    TMH0003         Tim Halvorsen           06-Jul-1981
1 !                     Remove version # checks on NML connect to allow communication
1 !                     between a 2.2 NCP and a 2.0 NML, which normally should not be
1 !                     allowed, but will be for compatibility after 2.2 release.
1 !
1 !             V02-002 LMK0001         Len Kawell              29-Sep-1980
1 !                     Change $DELMBX call to $DASSGN call.
1 !--
```

```
;   104           0103  1 %SBTTL  'Definitions'
;   105           0104  1
;   106           0105  1 !
;   107           0106  1 ! TABLE OF CONTENTS:
;   108           0107  1 !
;   109           0108  1
;   110           0109  1 FORWARD ROUTINE
;   111           0110  1         NCP$BLDLCB : NOVALUE,
;   112           0111  1         NCP$OPENLINK : NOVALUE,
;   113           0112  1         NCP$SIGNETERR : NOVALUE,
;   114           0113  1         NCP$CLOSELINK : NOVALUE,
;   115           0114  1         NCP$SENDMSG : NOVALUE,
;   116           0115  1         STORE_RESPONSE: NOVALUE,
;   117           0116  1         NCP$READRSP,
;   118           0117  1         NCP$TABLESEARCH
;   119           0118  1         ;               !
;   120           0119  1
;   121           0120  1 !
;   122           0121  1 ! INCLUDE FILES:
;   123           0122  1 !
;   124           0123  1
;   125           0124  1         LIBRARY 'SYS$LIBRARY:STARLET.L32';
;   126           0125  1         LIBRARY 'OBJ$:NMALIBRY.L32';
;   127           0126  1         LIBRARY 'OBJ$:NCPLIBRY.L32';
;   128           0127  1
;   129           0128  1 !
;   130           0129  1 ! MACROS:
;   131           0130  1 !
;   132           0131  1
;   133           0132  1 !
;   134           0133  1 ! EQUATED SYMBOLS:
;   135           0134  1 !
;   136           0135  1
;   137           0136  1 !
;   138           0137  1 !       Trailing portion of the Network Connect Block (NCB)
;   139           0138  1 !
;   140           0139  1 BIND
;   141           0140  1
;   142         P 0141  1         NCP$Q_OBJSPEC = ASCID ('::"19=/', %CHAR(0,0),
;   143         P 0142  1 !       NCP$Q_OBJSPEC = ASCID ('::"0=NML/', %CHAR(0,0),
;   144         P 0143  1                                 %CHAR(3, NCP$C_VRS, NCP$C_ECO, NCP$C_UECO),
;   145         P 0144  1                                 '"'
;   146           0145  2                                 )
;   147           0146  1         ;
;   148           0147  1
```

```
!
! OWN STORAGE:
!

!
!	Mailbox and Response buffers
!

GLOBAL
	NCP$GT_MBXBFR : VECTOR [NCP$C_MBXSIZ, BYTE],
	NCP$GT_RSPBFR : VECTOR [NCP$C_RSPSIZ, BYTE]
	;

!
!	Data to maintain the link control blocks for the executor
!

GLOBAL
	NCP$GT_EXECLCB : BBLOCK [LCB$C_SIZE],
	NCP$GT_TELLLCB : BBLOCK [LCB$C_SIZE],

	NCP$GL_OLDLCB,
	NCP$GL_EXELCB
	;

OWN
	NML_RESP_QUEUE: VECTOR [2]		! Local NML response queue header
		INITIAL(NML_RESP_QUEUE,NML_RESP_QUEUE);

!
! EXTERNAL REFERENCES:
!

EXTERNAL
	NCP$GL_FNC_CODE,			! Function code for command message
	NCP$GL_ENTITY;				! Entity number for command message

EXTERNAL ROUTINE
	NML$INITIALIZE: NOVALUE,		! Initialize NML sharable image
	NML$PROCESS_NICE: NOVALUE,		! Process a single NICE message
	NML$TERMINATE: NOVALUE,			! Terminate NML sharable image
	LIB$GET_VM,				! Allocate dynamic memory
	LIB$FREE_VM,				! Deallocate dynamic memory
	NCP$FORMATPARM : NOVALUE;		! Format a parameter as text
```

```
1 %SBTTL  'ACT$VRB_TELL  Process TELL Verb'
1 GLOBAL ROUTINE ACT$VRB_TELL  =  !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         Action routine to setup an executor node for one command.
1 !         Current executor LCB is saved and a newone is setup.
1 !         A link is opened to the new executor node.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         NCP$GL_OLDLCB             Save the current executor lcb
1 !         NCP$GL_EXELCB             The current executor lcb
1 !         NCP$GT_TELLLCB            LCB to use for tell
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NCP$GT_TELLLCB            Link opened
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !         Success or error signaled
1 !
1 ! SIDE EFFECTS:
1 !
1 !         NONE
1 !
1 !--
1
2     BEGIN
2
2     NCP$GL_OLDLCB = .NCP$GL_EXELCB;        ! Save the current executor
2     NCP$GL_EXELCB = NCP$GT_TELLLCB;        ! Set the new one
2     NCP$BLDLCB (.NCP$GL_EXELCB);           ! Build the link control block
2     NCP$OPENLINK (.NCP$GL_EXELCB);         ! Open the link
2     RETURN SUCCESS                         ! Always succeed for action
2
1     END;


                                                                             .TITLE  NCPNETIO Network I/O Routines
                                                                             .IDENT  \V04-000\

                                                                             .PSECT  $PLIT$,NOWRT,NOEXE,2

       00  00  04  03  00  00  2F  3D  39  31  22  3A  3A  00000 P.AAB:    .ASCII  \::"19=/\<0><0><3><4><0><0>                  :
00  22 20  20  20  20  20  20  20  20  20  20  20  20  20  0000D           .ASCII  \             "\<0>                          :
                                                 0000001B  0001C P.AAA:    .LONG   27                                           :
                                                 00000000' 00020           .ADDRESS P.AAB                                       :

                                                                             .PSECT  $OWN$,NOEXE,2
```

```
                                         00000000' 00000000' 00000 NML_RESP_QUEUE:
                                                                           .ADDRESS NML_RESP_QUEUE, NML_RESP_QUEUE                 ;

                                                                           .PSECT  $GLOBAL$,NOEXE,2

                                                             00000 NCP$GT_MBXBFR::
                                                                           .BLKB   40
                                                             00028 NCP$GT_RSPBFR::
                                                                           .BLKB   1000
                                                             00410 NCP$GT_EXECLCB::
                                                                           .BLKB   118
                                                             00486         .BLKB   2
                                                             00488 NCP$GT_TELLLCB::
                                                                           .BLKB   118
                                                             004FE         .BLKB   2
                                                             00500 NCP$GL_OLDLCB::
                                                                           .BLKB   4
                                                             00504 NCP$GL_EXELCB::
                                                                           .BLKB   4

                                                                   NCP$Q_OBJSPEC=      P.AAA
                                                                           .EXTRN  NCP$GL_FNC_CODE
                                                                           .EXTRN  NCP$GL_ENTITY, NML$INITIALIZE
                                                                           .EXTRN  NML$PROCESS_NICE
                                                                           .EXTRN  NML$TERMINATE, LIB$GET_VM
                                                                           .EXTRN  LIB$FREE_VM, NCP$FORMATPARM

                                                                           .PSECT  $CODE$,NOWRT,2

                                                        0004 00000         .ENTRY  ACT$VRB_TELL, Save R2                            ; 0194
                                       52 00000000'   00  9E 00002         MOVAB   NCP$GL_EXELCB, R2                                ;
                                   FC  A2             62  D0 00009         MOVL    NCP$GL_EXELCB, NCP$GL_OLDLCB                     ; 0230
                                       62        84   A2  9E 0000D         MOVAB   NCP$GT_TELLLCB, NCP$GL_EXELCB                    ; 0231
                                                      62  DD 00011         PUSHL   NCP$GL_EXELCB                                    ; 0232
                            00000000V  00             01  FB 00013         CALLS   #1, NCP$BLDLCB                                   ;
                                                      62  DD 0001A         PUSHL   NCP$GL_EXELCB                                    ; 0233
                            00000000V  00             01  FB 0001C         CALLS   #1, NCP$OPENLINK                                 ;
                                       50             01  D0 00023         MOVL    #1, R0                                           ; 0234
                                                          04 00026         RET                                                      ; 0236

; Routine Size:  39 bytes,    Routine Base:  $CODE$ + 0000
```

```
1
1 %SBTTL  'NCP$UNTELL  Remove the TELL Executor Node'
1 GLOBAL ROUTINE NCP$UNTELL :NOVALUE =     !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !        If the last command had a TELL prefix, the link to the temporary
1 !        executor is broken and the previous executor node is restored.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !        NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !        NCP$GL_OLDLCB            Pointer to previous executor LCB
1 !        NCP$GL_EXELCB            Tell executor LCB
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !        NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !        NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !        NONE
1 !
1 !--
1
2     BEGIN
2
2     IF .NCP$GL_OLDLCB NEQ 0              ! Is there a TELL outstanding?
2     THEN
3         BEGIN
3         NCP$CLOSELINK (.NCP$GL_EXELCB); ! Close the link to William TELL
3         NCP$GL_EXELCB = .NCP$GL_OLDLCB; ! Restore the old link
3         NCP$GL_OLDLCB = 0               ! There is no William TELL now
3         END
2     ;
2     RETURN
2
1     END;
```

```
                         0004 00000        .ENTRY  NCP$UNTELL, Save R2                    ; 0239
        52 00000000'  00  9E 00002         MOVAB   NCP$GL_OLDLCB, R2                      ;
                      62  D5 00009         TSTL    NCP$GL_OLDLCB                          ; 0273
                      10  13 0000B         BEQL    1$                                     ;
                 04   A2  DD 0000D         PUSHL   NCP$GL_EXELCB                          ; 0276
```

```
                         00000000V  00          01  FB 00010         CALLS   #1, NCP$CLOSELINK                          ;
                                04  A2          62  D0 00017         MOVL    NCP$GL_OLDLCB, NCP$GL_EXELCB               ; 0277
                                                62  D4 0001B         CLRL    NCP$GL_OLDLCB                              ; 0278
                                                    04 0001D 1$:     RET                                                ; 0283

; Routine Size:  30 bytes,    Routine Base:  $CODE$ + 0027
;   289          0284   1 %SBTTL  'ACT$VRB_SETEXEC  Establish the Executor Node'
;   290          0285   1 GLOBAL ROUTINE ACT$VRB_SETEXEC  =          !
;   291          0286   1
;   292          0287   1 !++
;   293          0288   1 ! FUNCTIONAL DESCRIPTION:
;   294          0289   1 !
;   295          0290   1 !       This is an action routine to establish a link to an executor node.
;   296          0291   1 !       Any previous link to an executor is broken.
;   297          0292   1 !       The LCB is built from data left by the parse and a link is
;   298          0293   1 !       opened.
;   299          0294   1 !
;   300          0295   1 ! FORMAL PARAMETERS:
;   301          0296   1 !
;   302          0297   1 !       NONE
;   303          0298   1 !
;   304          0299   1 ! IMPLICIT INPUTS:
;   305          0300   1 !
;   306          0301   1 !       NCP$GL_OLDLCB          Pointer to the lcb for the exec if tell active
;   307          0302   1 !       NCP$GL_EXELCB          Pointer to the lcb for the exec
;   308          0303   1 !       NCP$GT_EXECLCB         LCB to be used by the exec
;   309          0304   1 !
;   310          0305   1 ! IMPLICIT OUTPUTS:
;   311          0306   1 !
;   312          0307   1 !       NONE
;   313          0308   1 !
;   314          0309   1 ! ROUTINE VALUE:
;   315          0310   1 ! COMPLETION CODES:
;   316          0311   1 !
;   317          0312   1 !       Success or an error signaled by called routine
;   318          0313   1 !
;   319          0314   1 ! SIDE EFFECTS:
;   320          0315   1 !
;   321          0316   1 !       NONE
;   322          0317   1 !
;   323          0318   1 !--
;   324          0319   1
;   325          0320   2     BEGIN
;   326          0321   2
;   327          0322   2     NCP$GL_OLDLCB = 0;                   ! There is no tell active
;   328          0323   2     NCP$GL_EXELCB = NCP$GT_EXECLCB;      ! The executor data pointer is here
;   329          0324   2     NCP$CLOSELINK (.NCP$GL_EXELCB);      ! Close any previous link
;   330          0325   2     NCP$BLDLCB (.NCP$GL_EXELCB);         ! Build the lcb for the new executor
;   331          0326   2     NCP$OPENLINK (.NCP$GL_EXELCB);       ! Open the link to the new NML
;   332          0327   2     RETURN SUCCESS                       ! Action routine is always successful
;   333          0328   2
;   334          0329   1     END;



                                     0004 00000          .ENTRY  ACT$VRB_SETEXEC, Save R2                              ; 0285
                  52 00000000'    00   9E 00002          MOVAB   NCP$GL_EXELCB, R2                                     ;
                            FC    A2   D4 00009          CLRL    NCP$GL_OLDLCB                                         ; 0322
                  62      FF0C    C2   9E 0000C          MOVAB   NCP$GT_EXECLCB, NCP$GL_EXELCB                         ; 0323
                                  62   DD 00011          PUSHL   NCP$GL_EXELCB                                         ; 0324
         00000000V  00            01   FB 00013          CALLS   #1, NCP$CLOSELINK                                     ;
```

```
                                           62  DD 0001A        PUSHL   NCP$GL_EXELCB                                  ; 0325
                        00000000V  00      01  FB 0001C        CALLS   #1, NCP$BLDLCB                                 ;
                                           62  DD 00023        PUSHL   NCP$GL_EXELCB                                  ; 0326
                        00000000V  00      01  FB 00025        CALLS   #1, NCP$OPENLINK                               ;
                                   50      01  D0 0002C        MOVL    #1, R0                                         ; 0327
                                               04 0002F        RET                                                    ; 0329

; Routine Size:  48 bytes,    Routine Base:  $CODE$ + 0045
;   336          0330   1 %SBTTL  'ACT$VRB_CLEXEC  Close Link to the Executor'
;   337          0331   1 GLOBAL ROUTINE ACT$VRB_CLEXEC  =         !
;   338          0332   1
;   339          0333   1 !++
;   340          0334   1 ! FUNCTIONAL DESCRIPTION:
;   341          0335   1 !
;   342          0336   1 !       This is an action routine which closes the link to the current
;   343          0337   1 !       executor and opens a link to NML on the local node.
;   344          0338   1 !       The local node is known as '::' so we use the obj spec only
;   345          0339   1 !       to open a link to NML.
;   346          0340   1 !
;   347          0341   1 ! FORMAL PARAMETERS:
;   348          0342   1 !
;   349          0343   1 !       NONE
;   350          0344   1 !
;   351          0345   1 ! IMPLICIT INPUTS:
;   352          0346   1 !
;   353          0347   1 !       NCP$GT_EXECLCB          LCB to be used for the executor
;   354          0348   1 !       NCP$GL_OLDLCB           Pointer to lcb for tell exec
;   355          0349   1 !       NCP$GL_EXELCB           Pointer to lcb for exec
;   356          0350   1 !
;   357          0351   1 ! IMPLICIT OUTPUTS:
;   358          0352   1 !
;   359          0353   1 !       NONE
;   360          0354   1 !
;   361          0355   1 ! ROUTINE VALUE:
;   362          0356   1 ! COMPLETION CODES:
;   363          0357   1 !
;   364          0358   1 !       Success or error signaled
;   365          0359   1 !
;   366          0360   1 ! SIDE EFFECTS:
;   367          0361   1 !
;   368          0362   1 !       NONE
;   369          0363   1 !
;   370          0364   1 !--
;   371          0365   1
;   372          0366   2     BEGIN
;   373          0367   2
;   374          0368   2     LOCAL
;   375          0369   2         LCB : REF BBLOCK [LCB$C_SIZE],  ! Address of the lcb to be used
;   376          0370   2         PTR                             ! General pointer
;   377          0371   2         ;
;   378          0372   2
;   379          0373   2     NCP$GL_OLDLCB = 0;                  ! No tell is active
;   380          0374   2
;   381          0375   2     LCB = NCP$GT_EXECLCB;               ! The lcb of interest
;   382          0376   2     NCP$GL_EXELCB = .LCB;               ! The widely used pointer to it
;   383          0377   2     NCP$CLOSELINK (.LCB);               ! Close it if its active
;   384          0378   2
;   385          0379   2
;   386          0380   2 !
;   387          0381   2 !       Set a pointer to the NCB and put the obj spec on.  The local node
;   388          0382   2 !       will be used since we are using no node name and :: only is always
;   389          0383   2 !       the local node.
;   390          0384   2 !       Note we are using no access control, so the default access will be
;   391          0385   2 !       used for the object
;   392          0386   2 !
;   393        0387  2
;   394        0388  2        LCB [LCB$L_NCBPTR] = LCB [LCB$T_NCB];
;   395        0389  2
;   396        0390  2        PTR = .LCB [LCB$L_NCBPTR];
;   397        0391  2
;   398        0392  2        PTR = CH$MOVE
;   399        0393  2            (
;   400        0394  2            .BBLOCK [NCP$Q_OBJSPEC, DSC$W_LENGTH],
;   401        0395  2            .BBLOCK [NCP$Q_OBJSPEC, DSC$A_POINTER],
;   402        0396  2            .PTR
;   403        0397  2            );
;   404        0398  2
;   405        0399  2        LCB [LCB$L_NCBCNT] = .PTR - LCB [LCB$T_NCB];
;   406        0400  2        LCB [LCB$B_STS] = 0;
;   407        0401  2
;   408        0402  2                                         ! Link will be opened on first write
;   409        0403  2        RETURN SUCCESS
;   410        0404  2
;   411        0405  1        END;



                                           01FC 00000         .ENTRY  ACT$VRB_CLEXEC, Save R2,R3,R4,R5,R6,R7,R8      ; 0331
                         58 00000000'   00   9E 00002         MOVAB   NCP$GL_OLDLCB, R8                               ;
                                        68   D4 00009         CLRL    NCP$GL_OLDLCB                                   ; 0373
                         56       FF10  C8   9E 0000B         MOVAB   NCP$GT_EXECLCB, LCB                             ; 0375
                   04    A8             56   D0 00010         MOVL    LCB, NCP$GL_EXELCB                              ; 0376
                                        56   DD 00014         PUSHL   LCB                                             ; 0377
             00000000V   00             01   FB 00016         CALLS   #1, NCP$CLOSELINK                               ;
                         57         12  A6   9E 0001D         MOVAB   18(R6), R7                                      ; 0388
                   0E    A6             57   D0 00021         MOVL    R7, 14(LCB)                                     ;
                         53         0E  A6   D0 00025         MOVL    14(LCB), PTR                                    ; 0390
                         50 00000000'   00   D0 00029         MOVL    NCP$Q_OBJSPEC+4, R0                             ; 0395
              63         60 00000000'   00   28 00030         MOVC3   NCP$Q_OBJSPEC, (R0), (PTR)                      ; 0396
        0A    A6         53             57   C3 00038         SUBL3   R7, PTR, 10(LCB)                                ; 0399
                                        66   94 0003D         CLRB    (LCB)                                           ; 0400
                         50             01   D0 0003F         MOVL    #1, R0                                          ; 0403
                                             04 00042         RET                                                     ; 0405

; Routine Size:  67 bytes,    Routine Base:  $CODE$ + 0075
```

```
1 %SBTTL  'NCP$BLDLCB  Build an Link Control Block'
1 ROUTINE NCP$BLDLCB (LCB) :NOVALUE =      !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine builds the contents of an LCB (link control block)
1 !       from information in left around by the parse.
1 !       The nodename which may be a logical name, is translated 10 times or
1 !       until it does not translate further, which ever is first.
1 !       If access control is provided with the node spec, it is appended to
1 !       the translation after any access control is stripped from the
1 !       translation.  If no access control is provided in the node spec,
1 !       it may be specified in the logical. The logical name cannot contain
1 !       ::.  The translation may or may not contain ::.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       LCB                 Address of the link control block
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       PDB$G_VRB_XID             Node spec string
1 !       ACT$GQ_ACCACC_DSC         Descriptors of access control
1 !       ACT$GQ_ACCPSW_DSC
1 !       ACT$GQ_ACCUSR_DSC
1 !       ACT$GL_XIDACC_Q           True for access control in node spec
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2   BEGIN
2
2   MAP
2       LCB : REF BBLOCK                ! Pointer to an link control block
2       ;
2
2   LITERAL
2       RSLSIZ = 64                     ! Size for tranlation buffer
2       ;
2
2   LOCAL
2       RSLBUF : VECTOR [RSLSIZ, BYTE], ! Translation buffer
2       RSLDSC : VECTOR [2],            ! Descriptor of buffer
2       RSBDSC : VECTOR [2],            ! Descriptor of whole buffer
```

```
;  470          0463  2           STATUS,                         ! Return status of translation
;  471          0464  2           ACCPTR,                         ! Pointer to original access control
;  472          0465  2           ACCCNT,                         ! Size of access control
;  473          0466  2           PTR,
;  474          0467  2           CTR
;  475          0468  2           ;
;  476          0469  2
;  477          0470  2       EXTERNAL LITERAL
;  478          0471  2           NCP$_INVACC                     ! Invalid access control signal
;  479          0472  2           ;
;  480          0473  2
;  481          0474  2       EXTERNAL
;  482          0475  2           ACT$GQ_ACCACC_DSC,              ! Descriptors for access control
;  483          0476  2           ACT$GQ_ACCPSW_DSC,
;  484          0477  2           ACT$GQ_ACCUSR_DSC,
;  485          0478  2           ACT$GL_XIDACC_Q,                ! Access control present in nodespec
;  486          0479  2           PDB$G_VRB_XID                   ! Nodespec counted string here
;  487          0480  2           ;
;  488          0481  2
;  489          0482  2
;  490          0483  2 !
;  491          0484  2 !         Obtain the node spec and strip trailing colons
;  492          0485  2 !
;  493          0486  2
;  494          0487  2       PTR = BBLOCK [PDB$G_VRB_XID, PDB$T_DATA];       ! Obtain node spec string
;  495          0488  2       CTR = CH$RCHAR_A (PTR);                         ! And its size
;  496          0489  2
;  497          0490  2       DECRA IDX FROM .PTR + .CTR - 1                  ! Strip off trailing :: to
;  498          0491  2                 TO .PTR                               ! Begin translation
;  499          0492  2       DO
;  500          0493  2           IF CH$RCHAR (.IDX) EQL ':'
;  501          0494  2           THEN CTR = .CTR - 1
;  502          0495  2           ELSE EXITLOOP
;  503          0496  2       ;
;  504          0497  2
;  505          0498  2       CH$MOVE (.CTR, .PTR, RSLBUF);                   ! Copy to result buffer
;  506          0499  2       RSLDSC [0] = .CTR;                              ! Build descriptor
;  507          0500  2       RSLDSC [1] = RSLBUF;
;  508          0501  2       RSBDSC [1] = RSLBUF;                            ! Describe whole buffer too
;  509          0502  2       RSBDSC [0] = RSLSIZ;
;  510          0503  2
;  511          0504  2       IF .ACT$GL_XIDACC_Q                             ! If Access control specified
;  512          0505  2       THEN                                            ! Strip it off before trans
;  513          0506  3           BEGIN
;  514          0507  3           ACCPTR = CH$FIND_CH (.CTR, .PTR, '"'); ! Find it
;  515          0508  3           RSLDSC [0] = .ACCPTR - .PTR;                ! Shorten descriptor
;  516          0509  3           ACCCNT = .CTR - .RSLDSC [0]                 ! Size of our access control
;  517          0510  3           END
;  518          0511  2       ;
;  519          0512  2
;  520          0513  2       DECRU IDX FROM 10 TO 1                          ! Translate logical 10 deep
;  521          0514  2       DO
;  522          0515  3           BEGIN
;  523        P 0516  3           STATUS = $TRNLOG                            ! Obtain one translation
;  524        P 0517  3               (
;  525        P 0518  3               LOGNAM = RSLDSC,                        ! Here is the name to trans
;  526        P 0519  3               RSLLEN = RSLDSC [0],                    ! Return the length here
;   527       P 0520  3              RSLBUF = RSBDSC                     ! Return the string here
;   528         0521  4              )
;   529         0522  3          ;
;   530         0523  3          IF  NOT .STATUS                         ! If any error
;   531         0524  3              OR
;   532         0525  3              .STATUS EQL SS$_NOTRAN              ! or no translation
;   533         0526  3          THEN EXITLOOP                           ! we are done
;   534         0527  3          END
;   535         0528  2      ;
;   536         0529  2
;   537         0530  2      IF .ACT$GL_XIDACC_Q                         ! If node spec had acc control
;   538         0531  2      THEN                                        ! Use as override
;   539         0532  3          BEGIN
;   540         0533  3          PTR = CH$FIND_CH (.RSLDSC [0], .RSLDSC [1], '"');
;   541         0534  3          IF CH$FAIL (.PTR)                       ! If no acc in logical
;   542         0535  3          THEN
;   543         0536  3              PTR = .RSLDSC [1] + .RSLDSC [0]     ! Add ours at end
;   544         0537  3          ;
;   545         0538  3          PTR = CH$MOVE (.ACCCNT, .ACCPTR, .PTR); ! Add our acc ctl at end of
;   546         0539  3          RSLDSC [0] = .PTR - .RSLDSC [1]         ! translation
;   547         0540  3          END
;   548         0541  2      ;
;   549         0542  2
;   550         0543  2      PTR = LCB [LCB$T_NCB];                      ! Set pointer to start
;   551         0544  2      CH$MOVE                                     ! Copy node spec to lcb
;   552         0545  2          (
;   553         0546  2          .RSLDSC [0],
;   554         0547  2          .RSLDSC [1],
;   555         0548  2          .PTR
;   556         0549  2          );
;   557         0550  2      CTR = .RSLDSC [0];                          ! Set the counter for it
;   558         0551  2
;   559         0552  2      DECRA IDX FROM .PTR + .CTR - 1              ! Strip the colons again
;   560         0553  2                TO .PTR                           ! just to be sure
;   561         0554  2      DO
;   562         0555  2          IF CH$RCHAR (.IDX) EQL ':'
;   563         0556  2          THEN CTR = .CTR - 1
;   564         0557  2          ELSE EXITLOOP
;   565         0558  2      ;
;   566         0559  2
;   567         0560  2 !
;   568         0561  2 !        Obtain the access control if its needed
;   569         0562  2 !
;   570         0563  2
;   571         0564  2      PTR = LCB [LCB$T_NCB] + .CTR;               ! Point to the copied string
;   572         0565  2
;   573         0566  2      IF .ACT$GL_XIDACC_Q                         ! Is there access control in
;   574         0567  2      THEN                                        ! The node spec?
;   575         0568  3          BEGIN
;   576         0569  3          IF      .ACT$GQ_ACCACC_DSC NEQ 0        ! If so, there must not be
;   577         0570  3                  OR                              ! Access control elsewhere
;   578         0571  3                  .ACT$GQ_ACCPSW_DSC NEQ 0
;   579         0572  3                  OR
;   580         0573  3                  .ACT$GQ_ACCUSR_DSC NEQ 0
;   581         0574  3          THEN
;   582         0575  3              SIGNAL_STOP (NCP$_INVACC)           ! Signal too much access ctl
;   583         0576  3          END
```

```
2       ELSE
3           BEGIN
3           IF .ACT$GQ_ACCUSR_DSC NEQ 0                 ! If not, use other access ctl
3           THEN
4               BEGIN                                   ! Look through the name we
4               ACCPTR = CH$FIND_CH (.CTR, LCB [LCB$T_NCB], '"'); ! for acc ctl
4               IF NOT CH$FAIL (.ACCPTR)
4               THEN
4                   PTR = .ACCPTR
4               ;
4               CH$WCHAR_A ('"', PTR);                  ! Put it in standard form
4               PTR = CH$MOVE
4                   (
4                   .BBLOCK [ACT$GQ_ACCUSR_DSC, DSC$W_LENGTH],
4                   .BBLOCK [ACT$GQ_ACCUSR_DSC, DSC$A_POINTER],
4                   .PTR
4                   );
4
4               IF .ACT$GQ_ACCPSW_DSC NEQ 0             ! A password??
4               THEN
5                   BEGIN
5                   CH$WCHAR_A (' ', PTR);
5                   PTR = CH$MOVE
5                       (
5                       .BBLOCK [ACT$GQ_ACCPSW_DSC, DSC$W_LENGTH],
5                       .BBLOCK [ACT$GQ_ACCPSW_DSC, DSC$A_POINTER],
5                       .PTR
5                       )
5                   END
4               ELSE
4                   SIGNAL_STOP (NCP$_INVACC)           ! If no password, not complete
4               ;
4
4               IF .ACT$GQ_ACCACC_DSC NEQ 0             ! An account??
4               THEN
5                   BEGIN
5                   CH$WCHAR_A (' ', PTR);
5                   PTR = CH$MOVE
5                       (
5                       .BBLOCK [ACT$GQ_ACCACC_DSC, DSC$W_LENGTH],
5                       .BBLOCK [ACT$GQ_ACCACC_DSC, DSC$A_POINTER],
5                       .PTR
5                       )
5                   END
4               ;
4
4               CH$WCHAR_A ('"', PTR);                  ! End the access control spec
4
4               END
3           END
2       ;
2   !
2   !       Copy the object connect specification to the end
2   !
2
2       PTR = CH$MOVE
```

```
;  641         0634  2         (
;  642         0635  2         .BBLOCK [NCP$Q_OBJSPEC, DSC$W_LENGTH],
;  643         0636  2         .BBLOCK [NCP$Q_OBJSPEC, DSC$A_POINTER],
;  644         0637  2         .PTR
;  645         0638  2         );
;  646         0639  2
;  647         0640  2 !
;  648         0641  2 !   Fill up the LCB pointers and status
;  649         0642  2 !
;  650         0643  2
;  651         0644  2     LCB [LCB$L_NCBCNT] = .PTR - LCB [LCB$T_NCB];
;  652         0645  2     LCB [LCB$L_NCBPTR] = LCB [LCB$T_NCB];
;  653         0646  2     LCB [LCB$B_STS] = 0;
;  654         0647  2
;  655         0648  2     RETURN
;  656         0649  2
;  657         0650  1     END;



                                                                        .EXTRN  NCP$_INVACC, ACT$GQ_ACCACC_DSC
                                                                        .EXTRN  ACT$GQ_ACCPSW_DSC
                                                                        .EXTRN  ACT$GQ_ACCUSR_DSC
                                                                        .EXTRN  ACT$GL_XIDACC_Q
                                                                        .EXTRN  PDB$G_VRB_XID, SYS$TRNLOG

                                                     0FFC 00000 NCP$BLDLCB:
                                                                        .WORD  Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11       : 0407
                                      5B 00000000G  00 9E 00002         MOVAB  ACT$GQ_ACCACC_DSC, R11                     :
                                      5E        B0  AE 9E 00009         MOVAB  -80(SP), SP                                :
                                      57 00000000G  00 9E 0000D         MOVAB  PDB$G_VRB_XID+1, PTR                       : 0487
                                      58            87 9A 00014         MOVZBL (PTR)+, CTR                                : 0488
                        50            57            58 C1 00017         ADDL3  CTR, PTR, R0                               : 0490
                                                    07 11 0001B         BRB    2$                                         :
                                      3A            60 91 0001D 1$:     CMPB   (IDX), #58                                 : 0493
                                                    09 12 00020         BNEQ   3$                                         :
                                                    58 D7 00022         DECL   CTR                                        : 0494
                                                    50 D7 00024 2$:     DECL   IDX                                        : 0493
                                      57            50 D1 00026         CMPL   IDX, PTR                                   :
                                                    F2 1E 00029         BGEQU  1$                                         :
                   10   AE            67            58 28 0002B 3$:     MOVC3  CTR, (PTR), RSLBUF                         : 0498
                                 08   AE            58 D0 00030         MOVL   CTR, RSLDSC                                : 0499
                                 0C   AE        10  AE 9E 00034         MOVAB  RSLBUF, RSLDSC+4                           : 0500
                                 04   AE        10  AE 9E 00039         MOVAB  RSLBUF, RSBDSC+4                           : 0501
                                      6E        40  8F 9A 0003E         MOVZBL #64, RSBDSC                                : 0502
                                      15 00000000G  00 E9 00042         BLBC   ACT$GL_XIDACC_Q, 5$                        : 0504
                        67            58            22 3A 00049         LOCC   #34, CTR, (PTR)                            : 0507
                                                    02 12 0004D         BNEQ   4$                                         :
                                                    51 D4 0004F         CLRL   R1                                         :
                                      59            51 D0 00051 4$:     MOVL   R1, ACCPTR                                 :
                   08   AE            59            57 C3 00054         SUBL3  PTR, ACCPTR, RSLDSC                        : 0508
                        53            58        08  AE C3 00059         SUBL3  RSLDSC, CTR, ACCCNT                        : 0509
                                      52            0A D0 0005E 5$:     MOVL   #10, IDX                                   : 0513
                                                    7E 7C 00061 6$:     CLRQ   -(SP)                                      : 0521
                                                    7E D4 00063         CLRL   -(SP)                                      :
                                                0C  AE 9F 00065         PUSHAB RSBDSC                                     :
                                                18  AE 9F 00068         PUSHAB RSLDSC                                     :
```

```
                                                          1C   AE  9F 0006B         PUSHAB  RSLDSC                                          ;
                                    00000000G   00             06  FB 0006E         CALLS   #6, SYS$TRNLOG                                  ;
                                                0D             50  E9 00075         BLBC    STATUS, 7$                                      ; 0523
                                    00000629    8F             50  D1 00078         CMPL    STATUS, #1577                                   ; 0525
                                                               04  13 0007F         BEQL    7$                                              ;
                                                               52  D7 00081         DECL    IDX                                             ; 0523
                                                               DC  12 00083         BNEQ    6$                                              ;
                                                5A 00000000G   00  D0 00085 7$:     MOVL    ACT$GL_XIDACC_Q, R10                            ; 0530
                                                22             5A  E9 0008C         BLBC    R10, 10$                                        ;
                        0C    BE           08   AE             22  3A 0008F         LOCC    #34, RSLDSC, @RSLDSC+4                          ; 0533
                                                               02  12 00095         BNEQ    8$                                              ;
                                                               51  D4 00097         CLRL    R1                                              ;
                                                57             51  D0 00099 8$:     MOVL    R1, PTR                                         ;
                                                               06  12 0009C         BNEQ    9$                                              ; 0534
                              57           0C   AE        08   AE  C1 0009E         ADDL3   RSLDSC, RSLDSC+4, PTR                           ; 0536
                              67                69             53  28 000A4 9$:     MOVC3   ACCCNT, (ACCPTR), (PTR)                         ; 0538
                                                57             53  D0 000A8         MOVL    R3, PTR                                         ;
                        08    AE                57        0C   AE  C3 000AB         SUBL3   RSLDSC+4, PTR, RSLDSC                           ; 0539
                                                56        04   AC  D0 000B1 10$:    MOVL    LCB, R6                                         ; 0543
                                                57        12   A6  9E 000B5         MOVAB   18(R6), PTR                                     ;
                              67           0C   BE        08   AE  28 000B9         MOVC3   RSLDSC, @RSLDSC+4, (PTR)                        ; 0548
                                                58        08   AE  D0 000BF         MOVL    RSLDSC, CTR                                     ; 0550
                              50                57             58  C1 000C3         ADDL3   CTR, PTR, R0                                    ; 0552
                                                               07  11 000C7         BRB     12$                                             ;
                                                3A             60  91 000C9 11$:    CMPB    (IDX), #58                                      ; 0555
                                                               09  12 000CC         BNEQ    13$                                             ;
                                                               58  D7 000CE         DECL    CTR                                             ; 0556
                                                               50  D7 000D0 12$:    DECL    IDX                                             ; 0555
                                                57             50  D1 000D2         CMPL    IDX, PTR                                        ;
                                                               F2  1E 000D5         BGEQU   11$                                             ;
                                                57        12 A846  9E 000D7 13$:    MOVAB   18(CTR)[R6], PTR                                ; 0564
                                                23             5A  E9 000DC         BLBC    R10, 15$                                        ; 0566
                                                               6B  D5 000DF         TSTL    ACT$GQ_ACCACC_DSC                               ; 0569
                                                               10  12 000E1         BNEQ    14$                                             ;
                                                   00000000G   00  D5 000E3         TSTL    ACT$GQ_ACCPSW_DSC                               ; 0571
                                                               08  12 000E9         BNEQ    14$                                             ;
                                                   00000000G   00  D5 000EB         TSTL    ACT$GQ_ACCUSR_DSC                               ; 0573
                                                               7E  13 000F1         BEQL    21$                                             ;
                                                   00000000G   8F  DD 000F3 14$:    PUSHL   #NCP$_INVACC                                    ; 0575
                                    00000000G   00             01  FB 000F9         CALLS   #1, LIB$STOP                                    ;
                                                               6F  11 00100         BRB     21$                                             ; 0568
                                                   00000000G   00  D5 00102 15$:    TSTL    ACT$GQ_ACCUSR_DSC                               ; 0579
                                                               67  13 00108         BEQL    21$                                             ;
                        12    A6                58             22  3A 0010A         LOCC    #34, CTR, 18(R6)                                ; 0582
                                                               02  12 0010F         BNEQ    16$                                             ;
                                                               51  D4 00111         CLRL    R1                                              ;
                                                59             51  D0 00113 16$:    MOVL    R1, ACCPTR                                      ;
                                                               03  13 00116         BEQL    17$                                             ; 0583
                                                57             59  D0 00118         MOVL    ACCPTR, PTR                                     ; 0585
                                                87             22  90 0011B 17$:    MOVB    #34, (PTR)+                                     ; 0587
                                                50 00000000G   00  D0 0011E         MOVL    ACT$GQ_ACCUSR_DSC+4, R0                         ; 0591
                              67                60 00000000G   00  28 00125         MOVC3   ACT$GQ_ACCUSR_DSC, (R0), (PTR)                  ; 0592
                                                57             53  D0 0012D         MOVL    R3, PTR                                         ;
                                                   00000000G   00  D5 00130         TSTL    ACT$GQ_ACCPSW_DSC                               ; 0595
                                                               17  13 00136         BEQL    18$                                             ;
                                                87             20  90 00138         MOVB    #32, (PTR)+                                     ; 0598
                                                50 00000000G   00  D0 0013B         MOVL    ACT$GQ_ACCPSW_DSC+4, R0                         ; 0602
```

```
                      67              60 00000000G  00   28 00142          MOVC3   ACT$GQ_ACCPSW_DSC, (R0), (PTR)                 ; 0603
                                      57                 53   D0 0014A          MOVL    R3, PTR                                        ;
                                                    0D   11 0014D          BRB     19$                                            ; 0599
                                         00000000G  8F   DD 0014F  18$:    PUSHL   #NCP$_INVACC                                   ; 0607
                         00000000G    00            01   FB 00155          CALLS   #1, LIB$STOP                                   ;
                                                    6B   D5 0015C  19$:    TSTL    ACT$GQ_ACCACC_DSC                              ; 0610
                                                    0E   13 0015E          BEQL    20$                                            ;
                                      87            20   90 00160          MOVB    #32, (PTR)+                                    ; 0613
                                      50        04  AB   D0 00163          MOVL    ACT$GQ_ACCACC_DSC+4, R0                        ; 0617
                      67              60            6B   28 00167          MOVC3   ACT$GQ_ACCACC_DSC, (R0), (PTR)                 ; 0618
                                      57            53   D0 0016B          MOVL    R3, PTR                                        ;
                                      87            22   90 0016E  20$:    MOVB    #34, (PTR)+                                    ; 0623
                                      50 00000000'  00   D0 00171  21$:    MOVL    NCP$Q_OBJSPEC+4, R0                            ; 0636
                      67              60 00000000'  00   28 00178          MOVC3   NCP$Q_OBJSPEC, (R0), (PTR)                     ; 0637
                                      57            53   D0 00180          MOVL    R3, PTR                                        ;
                                      50        12  A6   9E 00183          MOVAB   18(R6), R0                                     ; 0644
                  0A  A6              57            50   C3 00187          SUBL3   R0, PTR, 10(R6)                                ;
                                0E    A6        12  A6   9E 0018C          MOVAB   18(R6), 14(R6)                                 ; 0645
                                                    66   94 00191          CLRB    (R6)                                           ; 0646
                                                         04 00193          RET                                                    ; 0650

; Routine Size:  404 bytes,    Routine Base:  $CODE$ + 00B8
```

```
1 %SBTTL  'NCP$OPENLINK  Open a link to NML'
1 GLOBAL ROUTINE NCP$OPENLINK (LCB) :NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine opens a link to NML given an LCB address and
1 !       verifies the connect data to determine if NML is phase II or
1 !       phase III.  The lcb already contains the NCB built in a previous
1 !       step.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       LCB             Address of the LCB to use
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE  errors signaled
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     LITERAL
2         MBXSIZ = 10                       ! Max size of mailbox name
2         ;
2
2     MAP
2         LCB : REF BBLOCK                  ! The link control block
2         ;
2
2     LOCAL
2         MBXBUF : VECTOR [MBXSIZ, BYTE],   ! Buffer to build mailbox name
2         MBXLST : VECTOR [2],              ! FAO list for mailbox name
2         MBXDSC : VECTOR [2],              ! Descriptor of mailbox name buffer
2         IOSB : BBLOCK [8],                ! IO status block
2         STATUS,                           ! Return status
2         PTR,                              ! General pointer
2         CTR                               ! General counter
2         ;
2
2     OWN
2         CHNCHAR : BBLOCK [DIB$K_LENGTH] ! Channel characteristics
2         ;
```

```
    EXTERNAL LITERAL
        NCP$_CONNEC,                        ! Connect errors
        NCP$_UNSVRS                         ! Unsupported version of nml
        ;

    LCB [LCB$W_MBXCHN] = 0;                 ! Make the channels zero
    LCB [LCB$W_CHAN] = 0;                   ! to indicate they are not here

    LCB [LCB$B_STS] = TRUE;                 ! This lcb is now open

    LCB [LCB$B_PH2] = FALSE;                ! Assume Phase III

    CH$FILL(0, 3, LCB [LCB$B_NMLVERS]); ! Preset NML version to null

!
!       If we are going to communicate with the NML on the local node,
!       and there is no access control string, then establish communications
!       with the sharable version of NML linked with this program, rather
!       than starting up another NML process on this node.
!

    IF CH$RCHAR(.LCB [LCB$L_NCBPTR]) EQL ':'
    THEN
        BEGIN
        NML$INITIALIZE();                   ! Initialize NICE processor
        CH$MOVE(3, UPLIT BYTE(NCP$C_VRS, NCP$C_ECO, NCP$C_UECO),
                LCB [LCB$B_NMLVERS]);   ! Assume NMLSHR is same as our version
        RETURN;                             ! Return successfully
        END;
!
!       We are about to do a non-transparent connect, so first
!       we must create a mailbox.
!

    STATUS = $CREMBX
            (
            CHAN = LCB [LCB$W_MBXCHN],
            MAXMSG = 64,
            BUFQUO = 256,
            PROMSK = %X'FF00'               ! own-sys=rwed
            );
    NCP$SIGNETERR (NCP$_CONNEC, .STATUS, 0);    ! Signal the error

    STATUS = $GETCHN                        ! Obtain the mailbox name
            (
            CHAN = .LCB [LCB$W_MBXCHN],
            PRIBUF = UPLIT (DIB$K_LENGTH, CHNCHAR)
            );
    NCP$SIGNETERR (NCP$_CONNEC, .STATUS, 0);    ! Report an error

    PTR = .CHNCHAR [DIB$W_DEVNAMOFF];       ! Offset to the name
    IF .PTR EQL 0                           ! Zero means missing
    THEN                                    ! No name, so we die here
        NCP$SIGNETERR (NCP$_CONNEC, SS$_IVCHAN, 0)
    ;
```

```
;   773           0765  2       MBXLST [0] = CHNCHAR + .PTR;          ! Data list has pointer to the name
;   774           0766  2       MBXLST [1] = .CHNCHAR [DIB$W_UNIT];   ! The unit number to convert
;   775           0767  2       MBXDSC [0] = MBXSIZ;                  ! Build descriptor of buffer, size and
;   776           0768  2       MBXDSC [1] = MBXBUF;                  ! Address of the buffer
;   777           0769  2
;   778         P 0770  2       $FAOL                                 ! Build the whole mailbox name
;   779         P 0771  2           (
;   780         P 0772  2           CTRSTR = ASCID ('_!AC!UW:'),      ! The name and unit _MBAnnn:
;   781         P 0773  2           OUTLEN = MBXDSC [0],              ! Length goes back in descriptor
;   782         P 0774  2           OUTBUF = MBXDSC,                  ! Descriptor is here
;   783         P 0775  2           PRMLST = MBXLST                   ! Data list is here
;   784           0776  2           );
;   785           0777  2
;   786         P 0778  2       STATUS = $ASSIGN                      ! Assign a channel to the network
;   787         P 0779  2               (
;   788         P 0780  2               DEVNAM = ASCID ('_NET:'),     ! General device for network
;   789         P 0781  2               CHAN = LCB [LCB$W_CHAN],      ! Place to put channel number
;   790         P 0782  2               MBXNAM = MBXDSC               ! Name we built with FAO
;   791           0783  2               );
;   792           0784  2       NCP$SIGNETERR (NCP$_CONNEC, .STATUS, 0); ! Report an error
;   793           0785  2
;   794         P 0786  2       STATUS = $QIOW                        ! Create a logical link to NML
;   795         P 0787  2               (
;   796         P 0788  2               CHAN = .LCB [LCB$W_CHAN],     ! Use network channel
;   797         P 0789  2               FUNC = IO$_ACCESS,            ! ACP function
;   798         P 0790  2               IOSB = IOSB,                  ! Status here
;   799         P 0791  2               P2 = LCB [LCB$L_NCBCNT]       ! This is the NCB descriptor
;   800           0792  2               );
;   801           0793  2       NCP$SIGNETERR (NCP$_CONNEC, .STATUS, IOSB); ! An error
;   802           0794  2
;   803         P 0795  2       STATUS = $QIOW                        ! Read the connect data
;   804         P 0796  2               (
;   805         P 0797  2               CHAN = .LCB [LCB$W_MBXCHN],   ! Channel for mailbox
;   806         P 0798  2               FUNC = IO$_READVBLK,
;   807         P 0799  2               IOSB = IOSB,
;   808         P 0800  2               P1 = NCP$GT_MBXBFR,           ! Read data into mailbox buffer
;   809         P 0801  2               P2 = NCP$C_MBXSIZ
;   810           0802  2               );
;   811           0803  2       NCP$SIGNETERR (NCP$_CONNEC, .STATUS, IOSB);
;   812           0804  2
;   813           0805  2   !
;   814           0806  2   !       Validate the message and its returned optional data
;   815           0807  2   !
;   816           0808  2
;   817           0809  2       STATUS = .BBLOCK [NCP$GT_MBXBFR, 0,0,16,0];
;   818           0810  2       PTR = NCP$GT_MBXBFR + 4;
;   819           0811  2
;   820           0812  2       IF .STATUS NEQ MSG$_CONFIRM           ! It must be a connect confirm
;   821           0813  2       THEN SIGNAL_STOP (NCP$_CONNEC)        ! Otherwise blow away
;   822           0814  2       ;
;   823           0815  2
;   824           0816  2       CTR = .IOSB [2, 0, 16, 0] - 4;        ! Play games to look at the data
;   825           0817  2       CTR = .CTR - CH$RCHAR (.PTR) - 1;     ! Skip over the device name
;   826           0818  2       PTR = .PTR + CH$RCHAR (.PTR) + 1;
;   827           0819  2
;   828           0820  2       IF CH$RCHAR (.PTR) LEQ 0              ! Any data returned?
;   829           0821  2       THEN  LCB [LCB$B_PH2] = TRUE          ! No, its phase II
;  830        0822  2       ELSE                                 ! Yes, check the data
;  831        0823  3           BEGIN
;  832        0824  3           IF
;  833        0825  4               (CH$RCHAR (.PTR) EQL 3)      ! And its size
;  834        0826  3               AND
;  835        0827  4               (CH$GEQ                      ! Check that version is current or later
;  836        0828  4                   (
;  837        0829  4                   3, .PTR + 1,
;  838        0830  4                   3, UPLIT (BYTE (NCP$C_VRS, NCP$C_ECO, NCP$C_UECO) ),
;  839        0831  4                   0
;  840        0832  4                   )
;  841        0833  4                OR
;  842        0834  4                CH$EQL                      ! or the version is V2.0
;  843        0835  4                   (
;  844        0836  4                   3, .PTR + 1,
;  845        0837  4                   3, UPLIT (BYTE (2, 0, 0) ),
;  846        0838  4                   0
;  847        0839  4                   )
;  848        0840  4                OR
;  849        0841  4                CH$EQL                      ! or the version is V3.0
;  850        0842  4                   (
;  851        0843  4                   3, .PTR + 1,
;  852        0844  4                   3, UPLIT (BYTE (3, 0, 0) ),
;  853        0845  4                   0
;  854        0846  4                   ))
;  855        0847  3           THEN
;  856        0848  4               BEGIN
;  857        0849  4               CH$MOVE(3, .PTR+1, LCB [LCB$B_NMLVERS]); ! Save NML version #
;  858        0850  4               LCB [LCB$B_PH2] = FALSE;     ! Its not phase II but phase III
;  859        0851  4               END
;  860        0852  3           ELSE
;  861        0853  4               BEGIN                        ! Close the link and blow away
;  862        0854  4               NCP$CLOSELINK (.LCB);
;  863        0855  4               SIGNAL_STOP (NCP$_UNSVRS)    ! Back with not a supported version
;  864        0856  4               END                          ! of nml
;  865        0857  3           END
;  866        0858  2       ;
;  867        0859  2
;  868        0860  2       RETURN
;  869        0861  2
;  870        0862  1       END;


                                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                00  00  04  00024 P.AAC:   .BYTE    4, 0, 0                              ;
                                                            00027          .BLKB    1
                                                  00000074  00028 P.AAD:   .LONG    116                                  ;
                                                  00000000' 0002C          .ADDRESS CHNCHAR                              ;
                        3A  57  55  21  43  41  21  5F  00030 P.AAF:   .ASCII   \_!AC!UW:\                           ;
                                                  00000008  00038 P.AAE:   .LONG    8                                    ;
                                                  00000000' 0003C          .ADDRESS P.AAF                                ;
                        00  00  00  3A  54  45  4E  5F  00040 P.AAH:   .ASCII   \_NET:\<0><0><0>                     ;
                                                  00000005  00048 P.AAG:   .LONG    5                                    ;
                                                  00000000' 0004C          .ADDRESS P.AAH                                ;
                                                00  00  04  00050 P.AAI:   .BYTE    4, 0, 0                              ;
```

```
                                                  00053             .BLKB    1
                                    00   00   02  00054 P.AAJ:      .BYTE    2, 0, 0                                       ;
                                                  00057             .BLKB    1
                                    00   00   03  00058 P.AAK:      .BYTE    3, 0, 0                                       ;

                                                                    .PSECT   $OWN$,NOEXE,2

                                                  00008 CHNCHAR:.BLKB    116

                                                                    .EXTRN   NCP$_CONNEC, NCP$_UNSVRS
                                                                    .EXTRN   SYS$CREMBX, SYS$GETCHN
                                                                    .EXTRN   SYS$FAOL, SYS$ASSIGN
                                                                    .EXTRN   SYS$QIOW

                                                                    .PSECT   $CODE$,NOWRT,2

                                             0FFC 00000             .ENTRY   NCP$OPENLINK, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0652
                                                                             R10,R11                                       :
                               5B 00000000G   00   9E 00002         MOVAB    SYS$QIOW, R11                                 :
                               5A 00000000'   00   9E 00009         MOVAB    NCP$GT_MBXBFR, R10                            :
                               59 00000000'   00   9E 00010         MOVAB    CHNCHAR+14, R9                                :
                               58 00000000V   00   9E 00017         MOVAB    NCP$SIGNETERR, R8                             :
                               57 00000000G   8F   D0 0001E         MOVL     #NCP$_CONNEC, R7                              :
                               56 00000000'   00   9E 00025         MOVAB    P.AAC, R6                                     :
                               5E             24   C2 0002C         SUBL2    #36, SP                                       :
                               55        04   AC   D0 0002F         MOVL     LCB, R5                                       : 0714
                                         02   A5   D4 00033         CLRL     2(R5)                                         : 0715
                               65             01   B0 00036         MOVW     #1, (R5)                                      : 0717
  06    A5          18         00             00   F0 00039         INSV     #0, #0, #24, 6(R5)                            : 0721
                               3A        0E   B5   91 0003F         CMPB     @14(R5), #58                                  : 0730
                                              0E   12 00043         BNEQ     1$                                            :
                      00000000G 00            00   FB 00045         CALLS    #0, NML$INITIALIZE                            : 0733
  06    A5          18         00             66   F0 0004C         INSV     P.AAC, #0, #24, 6(R5)                         : 0735
                                                   04 00052         RET                                                    : 0732
                                              7E   7C 00053 1$:     CLRQ     -(SP)                                         : 0749
                               7E      FF00   8F   3C 00055         MOVZWL   #65280, -(SP)                                 :
                               7E      0100   8F   3C 0005A         MOVZWL   #256, -(SP)                                   :
                               7E        40   8F   9A 0005F         MOVZBL   #64, -(SP)                                    :
                                         04   A5   9F 00063         PUSHAB   4(R5)                                         :
                                              7E   D4 00066         CLRL     -(SP)                                         :
                      00000000G 00            07   FB 00068         CALLS    #7, SYS$CREMBX                                :
                               52             50   D0 0006F         MOVL     R0, STATUS                                    :
                                              7E   D4 00072         CLRL     -(SP)                                         : 0750
                                              52   DD 00074         PUSHL    STATUS                                        :
                                              57   DD 00076         PUSHL    R7                                            :
                               68             03   FB 00078         CALLS    #3, NCP$SIGNETERR                             :
                                              7E   7C 0007B         CLRQ     -(SP)                                         : 0756
                                         04   A6   9F 0007D         PUSHAB   P.AAD                                         :
                                              7E   D4 00080         CLRL     -(SP)                                         :
                               7E        04   A5   3C 00082         MOVZWL   4(R5), -(SP)                                  :
                      00000000G 00            05   FB 00086         CALLS    #5, SYS$GETCHN                                :
                               52             50   D0 0008D         MOVL     R0, STATUS                                    :
                                              7E   D4 00090         CLRL     -(SP)                                         : 0757
                                              52   DD 00092         PUSHL    STATUS                                        :
                                              57   DD 00094         PUSHL    R7                                            :
                               68             03   FB 00096         CALLS    #3, NCP$SIGNETERR                             :
                               54             69   3C 00099         MOVZWL   CHNCHAR+14, PTR                               : 0759
```

```
                                     0C  12 0009C           BNEQ    2$                              ; 0760
                                     7E  D4 0009E           CLRL    -(SP)                           ; 0762
                      7E      013C   8F  3C 000A0           MOVZWL  #316, -(SP)                     ;
                                     57  DD 000A5           PUSHL   R7                              ;
                      68             03  FB 000A7           CALLS   #3, NCP$SIGNETERR               ;
                10    AE       F2 A944   9E 000AA  2$:      MOVAB   CHNCHAR[PTR], MBXLST            ; 0765
                14    AE        FE   A9  3C 000B0           MOVZWL  CHNCHAR+12, MBXLST+4            ; 0766
                08    AE             0A  D0 000B5           MOVL    #10, MBXDSC                     ; 0767
                0C    AE        18   AE  9E 000B9           MOVAB   MBXBUF, MBXDSC+4                ; 0768
                                10   AE  9F 000BE           PUSHAB  MBXLST                          ; 0776
                                0C   AE  9F 000C1           PUSHAB  MBXDSC                          ;
                                10   AE  9F 000C4           PUSHAB  MBXDSC                          ;
                                14   A6  9F 000C7           PUSHAB  P.AAE                           ;
         00000000G    00             04  FB 000CA           CALLS   #4, SYS$FAOL                    ;
                                08   AE  9F 000D1           PUSHAB  MBXDSC                          ; 0783
                                     7E  D4 000D4           CLRL    -(SP)                           ;
                                02   A5  9F 000D6           PUSHAB  2(R5)                           ;
                                24   A6  9F 000D9           PUSHAB  P.AAG                           ;
         00000000G    00             04  FB 000DC           CALLS   #4, SYS$ASSIGN                  ;
                      52             50  D0 000E3           MOVL    R0, STATUS                      ;
                                     7E  D4 000E6           CLRL    -(SP)                           ; 0784
                                     52  DD 000E8           PUSHL   STATUS                          ;
                                     57  DD 000EA           PUSHL   R7                              ;
                      68             03  FB 000EC           CALLS   #3, NCP$SIGNETERR               ;
                                     7E  7C 000EF           CLRQ    -(SP)                           ; 0792
                                     7E  7C 000F1           CLRQ    -(SP)                           ;
                                0A   A5  9F 000F3           PUSHAB  10(R5)                          ;
                                     7E  7C 000F6           CLRQ    -(SP)                           ;
                                     7E  D4 000F8           CLRL    -(SP)                           ;
                                20   AE  9F 000FA           PUSHAB  IOSB                            ;
                                     32  DD 000FD           PUSHL   #50                             ;
                      7E        02   A5  3C 000FF           MOVZWL  2(R5), -(SP)                    ;
                                     7E  D4 00103           CLRL    -(SP)                           ;
                      6B             0C  FB 00105           CALLS   #12, SYS$QIOW                   ;
                      52             50  D0 00108           MOVL    R0, STATUS                      ;
                              4004   8F  BB 0010B           PUSHR   #^M<R2,SP>                      ; 0793
                                     57  DD 0010F           PUSHL   R7                              ;
                      68             03  FB 00111           CALLS   #3, NCP$SIGNETERR               ;
                                     7E  7C 00114           CLRQ    -(SP)                           ; 0802
                                     7E  7C 00116           CLRQ    -(SP)                           ;
                                     28  DD 00118           PUSHL   #40                             ;
                                     5A  DD 0011A           PUSHL   R10                             ;
                                     7E  7C 0011C           CLRQ    -(SP)                           ;
                                20   AE  9F 0011E           PUSHAB  IOSB                            ;
                                     31  DD 00121           PUSHL   #49                             ;
                      7E        04   A5  3C 00123           MOVZWL  4(R5), -(SP)                    ;
                                     7E  D4 00127           CLRL    -(SP)                           ;
                      6B             0C  FB 00129           CALLS   #12, SYS$QIOW                   ;
                      52             50  D0 0012C           MOVL    R0, STATUS                      ;
                              4004   8F  BB 0012F           PUSHR   #^M<R2,SP>                      ; 0803
                                     57  DD 00133           PUSHL   R7                              ;
                      68             03  FB 00135           CALLS   #3, NCP$SIGNETERR               ;
                      52             6A  3C 00138           MOVZWL  NCP$GT_MBXBFR, STATUS           ; 0809
                      54        04   AA  9E 0013B           MOVAB   NCP$GT_MBXBFR+4, PTR            ; 0810
                      31             52  D1 0013F           CMPL    STATUS, #49                     ; 0812
                                     09  13 00142           BEQL    3$                              ;
                                     57  DD 00144           PUSHL   R7                              ; 0813
```

```
                            00000000G  00            01  FB 00146         CALLS   #1, LIB$STOP                                ;
                                       50      02    AE  3C 0014D 3$:     MOVZWL  IOSB+2, CTR                                 ; 0816
                                       50            04  C2 00151         SUBL2   #4, CTR                                     ;
                                       51            64  9A 00154         MOVZBL  (PTR), R1                                   ; 0817
                         52            50            51  C3 00157         SUBL3   R1, CTR, R2                                 ;
                                       50      FF    A2  9E 0015B         MOVAB   -1(R2), CTR                                 ;
                                       54      01 A144   9E 0015F         MOVAB   1(R1)[PTR], PTR                             ; 0818
                                                     64  95 00164         TSTB    (PTR)                                       ; 0820
                                                     05  12 00166         BNEQ    4$                                          ;
                                  01   A5            01  90 00168         MOVB    #1, 1(R5)                                   ; 0821
                                                         04 0016C         RET                                                 ;
                                       03            64  91 0016D 4$:     CMPB    (PTR), #3                                   ; 0825
                                                     23  12 00170         BNEQ    6$                                          ;
                    2C   A6       01   A4            03  29 00172         CMPC3   #3, 1(PTR), P.AAI                           ; 0828
                                                     10  1E 00178         BGEQU   5$                                          ;
                    30   A6       01   A4            03  29 0017A         CMPC3   #3, 1(PTR), P.AAJ                           ; 0835
                                                     08  13 00180         BEQL    5$                                          ;
                    34   A6       01   A4            03  29 00182         CMPC3   #3, 1(PTR), P.AAK                           ; 0842
                                                     0B  12 00188         BNEQ    6$                                          ;
      06   A5            18            00      01    A4  F0 0018A 5$:     INSV    1(PTR), #0, #24, 6(R5)                      ; 0849
                                               01    A5  94 00191         CLRB    1(R5)                                       ; 0850
                                                         04 00194         RET                                                 ; 0823
                                                     55  DD 00195 6$:     PUSHL   R5                                          ; 0854
                            00000000V  00            01  FB 00197         CALLS   #1, NCP$CLOSELINK                           ;
                                          00000000G  8F  DD 0019E         PUSHL   #NCP$_UNSVRS                                ; 0855
                            00000000G  00            01  FB 001A4         CALLS   #1, LIB$STOP                                ;
                                                         04 001AB         RET                                                 ; 0862

; Routine Size:  428 bytes,    Routine Base:  $CODE$ + 024C
```

```
:   872         0863   1 %SBTTL  'NCP$SIGNETERR  Signal a Network Communication Error'
:   873         0864   1 GLOBAL ROUTINE NCP$SIGNETERR (CODE, STATUS, IOSB) :NOVALUE =     !
:   874         0865   1
:   875         0866   1 !++
:   876         0867   1 ! FUNCTIONAL DESCRIPTION:
:   877         0868   1 !
:   878         0869   1 !       This routine checks the results from a system service or QIO
:   879         0870   1 !       and signals an error with a subcode.  Both the service status
:   880         0871   1 !       and the status block status is checked.
:   881         0872   1 !
:   882         0873   1 !       If there is an error on the link, it is closed if it is open.
:   883         0874   1 !       This will cause the link to be reopened automatically if another
:   884         0875   1 !       command is done.
:   885         0876   1 !
:   886         0877   1 ! FORMAL PARAMETERS:
:   887         0878   1 !
:   888         0879   1 !       CODE            Value of the NCP code to signal
:   889         0880   1 !       STATUS          Value of the system service status
:   890         0881   1 !       IOSB            Address of the IOSB to check for error status
:   891         0882   1 !
:   892         0883   1 ! IMPLICIT INPUTS:
:   893         0884   1 !
:   894         0885   1 !       NONE
:   895         0886   1 !
:   896         0887   1 ! IMPLICIT OUTPUTS:
:   897         0888   1 !
:   898         0889   1 !       NONE
:   899         0890   1 !
:   900         0891   1 ! ROUTINE VALUE:
:   901         0892   1 ! COMPLETION CODES:
:   902         0893   1 !
:   903         0894   1 !       NONE  error signaled with additional status
:   904         0895   1 !
:   905         0896   1 ! SIDE EFFECTS:
:   906         0897   1 !
:   907         0898   1 !       NONE
:   908         0899   1 !
:   909         0900   1 !--
:   910         0901   1
:   911         0902   2     BEGIN
:   912         0903   2
:   913         0904   2     MAP
:   914         0905   2         IOSB : REF BBLOCK
:   915         0906   2         ;
:   916         0907   2
:   917         0908   2     LOCAL
:   918         0909   2         REPORT
:   919         0910   2         ;
:   920         0911   2
:   921         0912   3     IF  NOT (REPORT = .STATUS)                      ! Look at the primary status
:   922         0913   2         OR
:   923         0914   2         NOT
:   924         0915   3         (
:   925         0916   3             IF .IOSB EQL 0                          ! If there is no iosb
:   926         0917   3             THEN TRUE                               ! Always succeed
:   927         0918   4             ELSE (REPORT = .IOSB [0, 0, 16, 0] ) ! Or report the iosb error
:   928         0919   3         )
```

```
;  929          0920  2      THEN
;  930          0921  3          BEGIN
;  931          0922  3          NCP$CLOSELINK (.NCP$GL_EXELCB);            ! Close link to mark to reopen
;  932          0923  3          SIGNAL_STOP (.CODE, 0, .REPORT)            ! Signal the error
;  933          0924  3          END
;  934          0925  3
;  935          0926  1      END;



                                     0004 00000           .ENTRY  NCP$SIGNETERR, Save R2                                 ; 0864
                        52       08  AC   D0 00002           MOVL    STATUS, REPORT                                         ; 0912
                        0C           52   E9 00006           BLBC    REPORT, 1$
                                 0C  AC   D5 00009           TSTL    IOSB                                                   ; 0916
                                     22   13 0000C           BEQL    2$
                        52       0C  BC   3C 0000E           MOVZWL  @IOSB, REPORT                                          ; 0918
                        1B           52   E8 00012           BLBS    REPORT, 2$
                           00000000'  00   DD 00015 1$:       PUSHL   NCP$GL_EXELCB                                          ; 0922
             00000000V  00           01   FB 0001B           CALLS   #1, NCP$CLOSELINK
                                     52   DD 00022           PUSHL   REPORT                                                 ; 0923
                                     7E   D4 00024           CLRL    -(SP)
                                 04  AC   DD 00026           PUSHL   CODE
             00000000G  00           03   FB 00029           CALLS   #3, LIB$STOP
                                          04 00030 2$:       RET                                                            ; 0926

; Routine Size:  49 bytes,    Routine Base:  $CODE$ + 03F8
```

```
1 %SBTTL  'NCP$CLOSELINK  Close a Link Open in an LCB'
1 GLOBAL ROUTINE NCP$CLOSELINK (LCB) :NOVALUE =    !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine closes a logical link open in an LCB.
1 !       The LCB$B_STS byte is true for the link is open.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       LCB             Address of the lcb describing the link
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE  return always occurs, error signaled non-fatal
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       LCB : REF BBLOCK                ! Link control block
2       ;
2
2    LOCAL
2       STATUS                          ! Service status
2       ;
2
2    EXTERNAL LITERAL
2       NCP$_DISCON                     ! Disconnect error status
2       ;
2
2    IF NOT .LCB [LCB$B_STS]            ! If link not open, return
2    THEN RETURN
2    ;
2
2    LCB [LCB$B_STS] = FALSE;           ! Mark its not open
2
2    IF CH$RCHAR(.LCB [LCB$L_NCBPTR]) EQL ':'    ! If talking to sharable NML,
2    THEN
3       BEGIN
3       BUILTIN REMQUE;
3       LOCAL
```

```
;   994        0984  3                  length,
;   995        0985  3                  entry: REF VECTOR;
;   996        0986  3              NML$TERMINATE();                  ! Perform sharable NML cleanups
;   997        0987  3              WHILE NOT REMQUE(.nml_resp_queue [0], entry) ! For each response in queue,
;   998        0988  3              DO
;   999        0989  4                  BEGIN
;  1000        0990  4                  length = .entry [2] + 12;    ! Length of entry
;  1001        0991  4                  LIB$FREE_VM(length, entry); ! Deallocate the entry
;  1002        0992  3                  END;
;  1003        0993  3              RETURN;
;  1004        0994  2              END;
;  1005        0995  2
;  1006        0996  2          IF .LCB [LCB$W_CHAN] NEQ 0
;  1007        0997  2          THEN
;  1008        0998  3              BEGIN
;  1009      P 0999  3              STATUS = $DASSGN                  ! Deassign the channel to net
;  1010        1000  3                  (CHAN = .LCB [LCB$W_CHAN]);
;  1011        1001  3              IF NOT .STATUS                    ! and report an error if so
;  1012        1002  3              THEN SIGNAL (NCP$_DISCON, 0, .STATUS)
;  1013        1003  3              END
;  1014        1004  2          ;
;  1015        1005  2
;  1016        1006  2          IF .LCB [LCB$W_MBXCHN] NEQ 0
;  1017        1007  2          THEN
;  1018        1008  3              BEGIN
;  1019      P 1009  3              STATUS = $DASSGN                  ! Deassign mailbox channel, deleting it
;  1020        1010  3                  (CHAN = .LCB [LCB$W_MBXCHN]);
;  1021        1011  3              IF NOT .STATUS                    ! and report the error
;  1022        1012  3              THEN SIGNAL (NCP$_DISCON, 0, .STATUS)
;  1023        1013  3              END
;  1024        1014  2          ;
;  1025        1015  2
;  1026        1016  2          RETURN
;  1027        1017  2
;  1028        1018  1          END;


                                                                   .EXTRN  NCP$_DISCON, SYS$DASSGN

                                               007C 00000          .ENTRY  NCP$CLOSELINK, Save R2,R3,R4,R5,R6                ; 0928
                      56 00000000G      00   9E 00002          MOVAB   LIB$SIGNAL, R6                                       ;
                      55 00000000G      8F   D0 00009          MOVL    #NCP$_DISCON, R5                                     ;
                      54 00000000G      00   9E 00010          MOVAB   SYS$DASSGN, R4                                       ;
                      5E                08   C2 00017          SUBL2   #8, SP                                               ;
                      50        04      AC   D0 0001A          MOVL    LCB, R0                                              ; 0973
                      6D                60   E9 0001E          BLBC    (R0), 4$                                             ;
                                        60   94 00021          CLRB    (R0)                                                 ; 0977
                      3A        0E      B0   91 00023          CMPB    @14(R0), #58                                         ; 0979
                                        2B   12 00027          BNEQ    2$                                                   ;
            00000000G 00                00   FB 00029          CALLS   #0, NML$TERMINATE                                    ; 0986
                      50 00000000'      00   9E 00030  1$:     MOVAB   NML_RESP_QUEUE, R0                                   ; 0987
                      6E        00      B0   0F 00037          REMQUE  @0(R0), ENTRY                                        ;
                                        51   1D 0003B          BVS     4$                                                   ;
                      50                6E   D0 0003D          MOVL    ENTRY, R0                                            ; 0990
   04    AE      08   A0                0C   C1 00040          ADDL3   #12, 8(R0), LENGTH                                   ;
                                        5E   DD 00046          PUSHL   SP                                                   ; 0991
```

```
                             08    AE  9F 00048          PUSHAB  LENGTH                          ;
              00000000G  00        02  FB 0004B          CALLS   #2, LIB$FREE_VM                 ;
                                   DC  11 00052          BRB     1$                              ; 0987
                         52    04  AC  D0 00054 2$:      MOVL    LCB, R2                         ; 0996
                               02  A2  B5 00058          TSTW    2(R2)                           ;
                                   16  13 0005B          BEQL    3$                              ;
                         7E    02  A2  3C 0005D          MOVZWL  2(R2), -(SP)                    ; 1000
                         64        01  FB 00061          CALLS   #1, SYS$DASSGN                  ;
                         53        50  D0 00064          MOVL    R0, STATUS                      ;
                         09        53  E8 00067          BLBS    STATUS, 3$                      ; 1001
                                   53  DD 0006A          PUSHL   STATUS                          ; 1002
                                   7E  D4 0006C          CLRL    -(SP)                           ;
                                   55  DD 0006E          PUSHL   R5                              ;
                         66        03  FB 00070          CALLS   #3, LIB$SIGNAL                  ;
                               04  A2  B5 00073 3$:      TSTW    4(R2)                           ; 1006
                                   16  13 00076          BEQL    4$                              ;
                         7E    04  A2  3C 00078          MOVZWL  4(R2), -(SP)                    ; 1010
                         64        01  FB 0007C          CALLS   #1, SYS$DASSGN                  ;
                         53        50  D0 0007F          MOVL    R0, STATUS                      ;
                         09        53  E8 00082          BLBS    STATUS, 4$                      ; 1011
                                   53  DD 00085          PUSHL   STATUS                          ; 1012
                                   7E  D4 00087          CLRL    -(SP)                           ;
                                   55  DD 00089          PUSHL   R5                              ;
                         66        03  FB 0008B          CALLS   #3, LIB$SIGNAL                  ;
                                       04 0008E 4$:      RET                                     ; 1018

; Routine Size:  143 bytes,    Routine Base:  $CODE$ + 0429
```

```
%SBTTL  'NCP$SENDMSG  Send a Message to NML'
GLOBAL ROUTINE NCP$SENDMSG (LCB, LEN, BFR) :NOVALUE =   !

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine sends a message to the NML object over the link
!       described by the LCB argument.  The buffer is described by the
!       remaining arguments.  System service and IO errors are signalled.
!
! FORMAL PARAMETERS:
!
!       LCB             Address of the link control block
!       LEN             Value of the length of the message
!       BFR             Address of the message buffer
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       NONE
!
! SIDE EFFECTS:
!
!       NONE
!
!--

    BEGIN

    MAP
        LCB : REF BBLOCK                ! Link control block
        ;

    LOCAL
        STATUS,                         ! Service status
        IOSB : BBLOCK [8]               ! IO status block
        ;

    EXTERNAL LITERAL
        NCP$_NETIO                      ! Network comm error
        ;

    IF NOT .LCB [LCB$B_STS]             ! If link is not open
    THEN
        NCP$OPENLINK (.LCB);            ! Open the link to executor

    IF CH$RCHAR(.LCB [LCB$L_NCBPTR]) EQL ':'    ! If talking to sharable NML,
    THEN
        BEGIN
```

```
; 1087        1076  3           BUILTIN REMQUE;
; 1088        1077  3           LOCAL
; 1089        1078  3               length,
; 1090        1079  3               entry:      REF VECTOR,
; 1091        1080  3               msgdesc:    VECTOR [2];
; 1092        1081  3
; 1093        1082  3           WHILE NOT REMQUE(.nml_resp_queue [0], entry) ! For each response in queue,
; 1094        1083  3           DO
; 1095        1084  4               BEGIN
; 1096        1085  4               length = .entry [2] + 12;   ! Length of entry
; 1097        1086  4               LIB$FREE_VM(length, entry); ! Deallocate the entry
; 1098        1087  3               END;
; 1099        1088  3
; 1100        1089  3           msgdesc [0] = .len;                         ! Make descriptor of message
; 1101        1090  3           msgdesc [1] = .bfr;
; 1102        1091  3           NML$PROCESS_NICE(msgdesc,                   ! Call sharable NML with message
; 1103        1092  3                            store_response);           ! and store all the responses
; 1104        1093  3           RETURN;
; 1105        1094  2           END;
; 1106        1095  2
; 1107      P 1096  2       STATUS = $QIOW                          ! Write the message
; 1108      P 1097  2               (
; 1109      P 1098  2               CHAN = .LCB [LCB$W_CHAN],
; 1110      P 1099  2               FUNC = IO$_WRITEVBLK,
; 1111      P 1100  2               IOSB = IOSB,
; 1112      P 1101  2               P1 = .BFR,
; 1113      P 1102  2               P2 = .LEN
; 1114        1103  2               );
; 1115        1104  2       NCP$SIGNETERR (NCP$_NETIO, .STATUS, IOSB); ! Check and signal an error
; 1116        1105  2
; 1117        1106  2       RETURN
; 1118        1107  2
; 1119        1108  1       END;



                                                              .EXTRN  NCP$_NETIO

                                             0004 00000       .ENTRY  NCP$SENDMSG, Save R2                                   ; 1020
                    5E                  18   C2 00002         SUBL2   #24, SP
                    52          04      AC   D0 00005         MOVL    LCB, R2                                                ; 1069
                    07                  62   E8 00009         BLBS    (R2), 1$
                                        52   DD 0000C         PUSHL   R2                                                     ; 1071
          FD81      CF                  01   FB 0000E         CALLS   #1, NCP$OPENLINK
                    3A          0E      B2   91 00013 1$:     CMPB    @14(R2), #58                                           ; 1073
                                        3A   12 00017         BNEQ    4$
                    50 00000000'        00   9E 00019 2$:     MOVAB   NML_RESP_QUEUE, R0                                     ; 1082
                    6E          00      B0   0F 00020         REMQUE  @0(R0), ENTRY
                                        17   1D 00024         BVS     3$
                    50                  6E   D0 00026         MOVL    ENTRY, R0                                              ; 1085
    04      AE          08      A0      0C   C1 00029         ADDL3   #12, 8(R0), LENGTH
                                        5E   DD 0002F         PUSHL   SP                                                     ; 1086
                                08      AE   9F 00031         PUSHAB  LENGTH
        00000000G   00                  02   FB 00034         CALLS   #2, LIB$FREE_VM
                                        DC   11 0003B         BRB     2$                                                     ; 1082
            08      AE          08      AC   7D 0003D 3$:     MOVQ    LEN, MSGDESC                                           ; 1089
                        00000000V       00   9F 00042         PUSHAB  STORE_RESPONSE                                         ; 1091
```

```
                                        0C   AE  9F 00048           PUSHAB  MSGDESC                             ;
                      00000000G  00          02  FB 0004B           CALLS   #2, NML$PROCESS_NICE                ;
                                                 04 00052           RET                                         ; 1075
                                             7E  7C 00053 4$:       CLRQ    -(SP)                               ; 1103
                                             7E  7C 00055           CLRQ    -(SP)                               ;
                                        08   AC  DD 00057           PUSHL   LEN                                 ;
                                        0C   AC  DD 0005A           PUSHL   BFR                                 ;
                                             7E  7C 0005D           CLRQ    -(SP)                               ;
                                        30   AE  9F 0005F           PUSHAB  IOSB                                ;
                                             30  DD 00062           PUSHL   #48                                 ;
                                 50     04   AC  D0 00064           MOVL    LCB, R0                             ;
                                 7E     02   A0  3C 00068           MOVZWL  2(R0), -(SP)                        ;
                                             7E  D4 0006C           CLRL    -(SP)                               ;
                      00000000G  00          0C  FB 0006E           CALLS   #12, SYS$QIOW                       ;
                                        10   AE  9F 00075           PUSHAB  IOSB                                ; 1104
                                             50  DD 00078           PUSHL   STATUS                              ;
                                 00000000G   8F  DD 0007A           PUSHL   #NCP$_NETIO                         ;
                           FEBB  CF          03  FB 00080           CALLS   #3, NCP$SIGNETERR                   ;
                                                 04 00085           RET                                         ; 1108

; Routine Size:  134 bytes,    Routine Base:  $CODE$ + 04B8
```

```
1 %SBTTL  'STORE_RESPONSE  Store a response from sharable NML'
1 ROUTINE store_response (resp_desc): NOVALUE =
1
1 !++
1 !
1 !
1 !        This routine is called by NML$PROCESS_NICE for each response
1 !        that it generates as a result of processing a single NICE message.
1 !        All we do is store the response messages away in a queue in the
1 !        order in which they were generated, and de-queue them later when
1 !        we wish to "read" a response.
1 !
1 ! Inputs:
1 !
1 !        resp_desc = Address of descriptor of NICE response message
1 !
1 ! Outputs:
1 !
1 !        None
1 !--
1
2 BEGIN
2
2 BUILTIN INSQUE;
2
2 MAP
2     resp_desc:  REF BBLOCK;              ! Address of response descriptor
2
2 LOCAL
2     status,
2     length,                              ! Length of block containing response
2     entry:      REF VECTOR;              ! Address of block to contain response
2
2 length = .resp_desc [dsc$w_length] + 12; ! Add response length + overhead
2
2 status = LIB$GET_VM(length, entry);      ! Allocate dynamic memory
2
2 IF NOT .status                           ! If error detected,
2 THEN
2     SIGNAL_STOP(.status);                ! then signal fatal error
2
2 entry [2] = .resp_desc [dsc$w_length];   ! Store length of response message
2 CH$MOVE(.resp_desc [dsc$w_length],       ! Copy message to new block
2         .resp_desc [dsc$a_pointer],
2         entry [3]);
2
2 INSQUE(.entry, .nml_resp_queue [1]);     ! Insert at end of queue
2
1 END;
```

```
                              007C 00000 STORE_RESPONSE:
                                                .WORD   Save R2,R3,R4,R5,R6                          ; 1110
          5E              08  C2 00002          SUBL2   #8, SP                                       ;
          52      04      AC  D0 00005          MOVL    RESP_DESC, R2                                ; 1141
```

```
                            04   AE            62  3C 00009          MOVZWL   (R2), LENGTH                                    ;
                            04   AE            0C  C0 0000D          ADDL2    #12, LENGTH                                     ;
                                               5E  DD 00011          PUSHL    SP                                              ; 1143
                                        08     AE  9F 00013          PUSHAB   LENGTH                                          ;
                     00000000G  00             02  FB 00016          CALLS    #2, LIB$GET_VM                                  ;
                                09             50  E8 0001D          BLBS     STATUS, 1$                                      ; 1145
                                               50  DD 00020          PUSHL    STATUS                                          ; 1147
                     00000000G  00             01  FB 00022          CALLS    #1, LIB$STOP                                    ;
                                56             6E  D0 00029 1$:      MOVL     ENTRY, R6                                       ; 1149
                            08  A6             62  3C 0002C          MOVZWL   (R2), 8(R6)                                     ;
               0C    A6     04  B2             62  28 00030          MOVC3    (R2), @4(R2), 12(R6)                            ; 1152
                                50  00000000'  00  9E 00036          MOVAB    NML_RESP_QUEUE+4, R0                            ; 1154
                            00  B0             66  0E 0003D          INSQUE   (R6), @0(R0)                                    ;
                                               04 00041              RET                                                      ; 1156

; Routine Size:  66 bytes,    Routine Base:  $CODE$ + 053E
```

```
1 %SBTTL  'NCP$READRSP  Read and Decode an NML Response'
1 GLOBAL ROUTINE NCP$READRSP (LCB, LEN, BFR, SHO) =          !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine reads a message from NML and decodes it.
1 !       If the message is an error response, the error is signaled and
1 !       control does not return to the caller.
1 !       If the message is a data return or a done status, the message is
1 !       returned via LEN, BFR and the first byte is returned as the value of
1 !       the routine. LEN and BFR form a descriptor of the data beyond the
1 !       error status byte, detail and error message.  If the error status
1 !       is SUC, DON or MOR, and there is a detail or error message, an
1 !       error is signaled to print these but control returns normally to
1 !       the caller.
1 !
1 !       If an error contains data, it is assumed to be an entity for the
1 !       error and the entity code is formatted and included in the error
1 !       message.  Entity codes may also occur with success codes and in
1 !       this case the data is printed as an entity if the message is not
1 !       a show or list command, indicated by the SHO parameter.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       LCB             Address of link control block
1 !       LEN             Address for return of length of buffer
1 !       BFR             Address for return of address of buffer
1 !       SHO             True if the command is show or list
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NCP$GL_ENTITY   Entity number sent in original message
1 !                       (If negative, then system-specific entity)
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Value of first byte of message, or error signalled
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2   BEGIN
2
2   MAP
2       LCB : REF BBLOCK                ! Link control block
2       ;
2
2   LITERAL
```

```
2           RSPSIZ = 32,                      ! Size of response buffer required
2           DTLSIZ = 32,                      ! Size of detail buffer required
2           ENTSIZ = 32                       ! Size of entity code buffer
2           ;
2
2       LOCAL
2           STATUS,                           ! Service status return
2           OUTLEN,                           ! Length in a buffer
2           IOSB : BBLOCK [8],                ! QIO status
2           CTR,                              ! General temps
2           PTR,
2           CODE,
2           ENTITY,                           ! Entity number (negative if sys-specific)
2           RSP,                              ! Pointer for response text
2           COMMA,                            ! Pointer to separator before detail
2           DTL,                              ! Pointer for detail text
2           ERR,                              ! Pointer for error text
2           ENT,                              ! Pointer for entity code text
2           IDX,                              ! Index into tables
2           JUNK,                             ! Throw away temporary
2           DETAIL,                           ! Value of detail word
2           DTLTBL                            ! Address of detail table
2           ;
2
2       OWN
2           DTLBUF : VECTOR [DTLSIZ, BYTE],   ! Detail buffer
2           RSPBUF : VECTOR [RSPSIZ, BYTE],   ! Response buffer
2           ENTDSC : VECTOR [2],              ! Descriptor for string
2           ENTBUF : VECTOR [ENTSIZ, BYTE]    ! Entity string buffer
2           ;
2
2
2       EXTERNAL LITERAL
2           NCP$_NMLRSP,                      ! NML response message
2           NCP$_NETIO                        ! Network communication error
2           ;
2
2       EXTERNAL
2           NCP$GA_TBL_NMLSTS,                ! NML status return codes
2           NCP$GA_TBL_FOPDTL,                ! File operations detail codes
2           NCP$GA_TBL_NCEDTL,                ! Network communications detail codes
2           NCP$GA_TBL_VMSENTDTL,             ! Detail table of VMS specific entities
2           NCP$GA_TBL_ENTDTL,                ! Detail table of entities
2           NCP$GA_TBL_OPEDTL;                ! Detail table of operation failures
2
2       EXTERNAL ROUTINE
2           NCP$FAOSET        : NOVALUE,      ! Setup to convert entity
2           NCP$SHOENTITY     : NOVALUE,      ! Convert entity
2           NCP$FAOL          : NOVALUE       ! Convert fao string for entity
2           ;
2
2       .LEN = 0;                             ! Set callers data
2       .BFR = NCP$GT_RSPBFR;
2
2       IF CH$RCHAR(.LCB [LCB$L_NCBPTR]) EQL ':'     ! If talking to sharable NML,
2       THEN
3           BEGIN
```

```
    BUILTIN REMQUE;
    LOCAL
        length,
        entry:      REF VECTOR;
    IF REMQUE(.nml_resp_queue [0], entry)   ! De-queue next one.  If none,
    THEN
        SIGNAL_STOP(NCP$_NETIO, SS$_ABORT); ! signal fatal error
    ctr = .entry [2];                   ! Copy length of response
    ptr = ncp$gt_rspbfr;                ! Set address of buffer
    CH$MOVE(.ctr, entry [3], .ptr);     ! Copy response into buffer
    length = .ctr + 12;                 ! Set length of container block
    LIB$FREE_VM(length, entry);         ! Deallocate container block
    END
ELSE                                    ! Else, read response from logical link
    BEGIN
    STATUS = $QIOW                      ! Read the message from NML
                (
                CHAN = .LCB [LCB$W_CHAN],
                FUNC = IO$_READVBLK,
                IOSB = IOSB,
                P1 = NCP$GT_RSPBFR,
                P2 = NCP$C_RSPSIZ
                );
    NCP$SIGNETERR (NCP$_NETIO, .STATUS, IOSB);

    CTR = .IOSB [0, 16, 16, 0];                 ! Point and count into message
    PTR = NCP$GT_RSPBFR;
    END;

!
!   We need to set some defaults in case the message is bad
!

RSP = UPLIT (%ASCIC 'unrecognized'); ! Some default text for message
COMMA = UPLIT (%ASCIC '');
DTL = UPLIT (%ASCIC '');
ENT = UPLIT (%ASCIC '');
ERR = UPLIT (%ASCIC '');

IF .CTR EQL 0                           ! If message is short, signal now
THEN
    SIGNAL_STOP (NCP$_NMLRSP, 5, .RSP, .COMMA, .DTL, .ENT, .ERR)
;

CODE = .(.PTR) <0, 8, 1>;               ! First byte is a code

IF NOT NCP$TABLESEARCH                  ! Find the code text if possible
        (
        .CODE <0, 8, 0>,                ! Code byte
        NCP$GA_TBL_NMLSTS,              ! Table
        RSP                             ! Return address of counted string
        )

THEN
    BEGIN
    $FAO                                ! If not found, make some text
        (
P 3             ASCID ('management return # !SB'),
P 3             OUTLEN,
P 3             UPLIT (RSPSIZ-1, RSPBUF+1),
P 3             .CODE
  3             );
  3         RSPBUF [0] = .OUTLEN;               ! As a counted string
  3         RSP = RSPBUF                        ! Point to it
  3         END
  2     ;
  2
  2     DETAIL = -1;                            ! No detail yet
  2
  2     IF .CTR GEQ 3                           ! Is there a detail word
  2     THEN
  3         BEGIN
  3         DETAIL = .(.PTR+1) <0, 16, 1>;      ! Obtain the word
  3         IF .DETAIL NEQ -1                   ! Ignore value?
  3         THEN
  4             BEGIN                           ! Nope
  4             DTLTBL =                        ! Find a table to use
  5                 BEGIN
  5                 SELECTONE .CODE OF
  5                 SET
  5                 [NMA$C_STS_FOP, NMA$C_STS_FIO, NMA$C_STS_FCO] :
  5                         NCP$GA_TBL_FOPDTL       ! File io errors
  5                         ;
  5                 [NMA$C_STS_MLD, NMA$C_STS_MCF] :
  5                         NCP$GA_TBL_NCEDTL       ! Network io errors
  5                         ;
  5                 [NMA$C_STS_OPE] :
  5                         NCP$GA_TBL_OPEDTL       ! Operation failure
  5                         ;
  5                 [NMA$C_STS_CMP, NMA$C_STS_IDE, NMA$C_STS_STA] :
  5                                                 ! Errors with entities
  5                         IF .NCP$GL_ENTITY LSS 0 ! If system-specific entity
  5                         THEN
  5                                 NCP$GA_TBL_VMSENTDTL    ! VMS entities
  5                         ELSE
  5                                 NCP$GA_TBL_ENTDTL;      ! DNA entities
  5                 [OTHERWISE] :                   ! Details not valid
  6                         BEGIN
  6                         IF .DETAIL EQL 0        ! Zero is null detail here
  6                         THEN 1                  ! Null detail if not valid
  6                         ELSE 0                  ! But report non zero detail
  6                         END
  5                         ;
  5                 TES
  5                 END
  4             ;
  4
  4             IF  .CODE EQL NMA$C_STS_OPE         ! If operation failure
  4             AND
  5                 (.NCP$GL_ENTITY EQL             !  and entity is line
  5                   NMA$C_ENT_LIN
  5             OR
  5                  .NCP$GL_ENTITY EQL             !  or circuit
  5                   NMA$C_ENT_CIR)
```

```
5
4            THEN
5                BEGIN
5                   LOCAL
5                   PREBUF : VECTOR [40, BYTE],  ! Buffer for string to proceed
5                                                !  each detail message.
5                   PRELEN,                      ! Length of string to proceed
5                                                !  each detail message.
5                   LOCPTR;                      ! Local pointer
5
5                LOCPTR = PREBUF;                ! Init pointer into buffer
5
5 !
5 ! Build the string which will preceed the detail text so that each detail
5 ! string output will line-up under the error text.  For example:
5 !
5 !      %facility-L-ident, error text           ! Original error message
5 !
5 !      %facility-L-ident, error text<CR><LF>   ! Message with two detail
5 !      <    SPACES   >, detail text<CR><LF>    !  strings appended.
5 !      <    SPACES   >, detail text            !
5 !
6                PRELEN = ( CH$FIND_CH(.(.PTR+3),! Get the number of characters
6                           .PTR + 4, %C',') )   !  in the facility and ident
5                            - (.PTR + 4);       !  portion of error message
5
5                (.LOCPTR) <0, 16> = %X'0A0D';   ! Store <CR><LF> in buffer,
5                LOCPTR = CH$FILL( %C' ', .PRELEN, .LOCPTR + 2 ); ! some spaces,
5                (.LOCPTR) <0, 16> = %ASCII', '; !  and a ", "
5                PRELEN = .PRELEN + 4;           ! Length = length of facility
5                                                ! text plus <CR><LF> and ", "
5
5                LOCPTR = .PTR + 4 +             ! Point to end of original
5                          .(.PTR + 3) < 0, 8 >; !  error message text.
5
5                INCR INDEX FROM 0 TO 16 DO
6                    BEGIN
6                    IF .DETAIL < .INDEX, 1, 0 > ! If status or error bit is set,
6                    AND                         !  and it's in the table,
6                    NCP$TABLESEARCH (.INDEX, .DTLTBL, DTL)
6                    AND                         !  and there's room in the
6                    .PRELEN  + .(.DTL) < 0, 8 > !  response buffer.
6                    LEQ .PTR + NCP$C_RSPSIZ - .LOCPTR
6                    THEN
7                        BEGIN
7                        LOCPTR = CH$MOVE        ! Append the string which
7                                    (           !  preceeds each detail message
7                                    .PRELEN,    !  to the end of the error
7                                    PREBUF,     !  message
7                                    .LOCPTR     !
7                                    );          !
7
7
7                        LOCPTR = CH$MOVE        ! Append detail to end of the
7                                    (           !  error message
7                                    .(.DTL) <0,8>,
7                                    .DTL + 1,   !
7                                     .LOCPTR           !
7                                     );                !
6                         END;
5                    END;

5                 (.PTR + 3) < 0, 8 > =                 ! Update message length.
5                             .LOCPTR - .PTR - 4;       !
5                 CTR < 0, 8 > = .LOCPTR - .PTR;        ! Update counter.
5                 DTLTBL = 1;                           ! Indicate that we formatted it
5                 DTLBUF [0] = 0;                       ! Make sure we Don't print the
5                 DTL = DTLBUF;                         !  detail #

5                 END

4             ELSE
4             IF  .CODE EQL NMA$C_STS_PVA               ! Special details for these
4                 OR                                    ! Errors, its the parameter
4                 .CODE EQL NMA$C_STS_PLO               ! name
4                 OR
4                 .CODE EQL NMA$C_STS_PNA
4                 OR
4                 .CODE EQL NMA$C_STS_PTY
4                 OR
4                 .CODE EQL NMA$C_STS_PGP
4                 OR
4                 .CODE EQL NMA$C_STS_PMS
4             THEN
5                 BEGIN
5                 NCP$FORMATPARM                        ! Format the parameter name
5                     (
5                     .NCP$GL_ENTITY,                   ! Entity is here
5                     DETAIL,                           ! Parameter code is here
5                     TRUE,                             ! Give the name
5                     FALSE,                            ! Not the data
5                     UPLIT (DTLSIZ - 1, DTLBUF + 1), ! Describe the buffer
5                     OUTLEN,                           ! Length of text here
5                     JUNK                              ! Return pointer to throw away
5                     );
5                 DTLBUF [0] = .OUTLEN;                 ! Set length of counted string
5                 DTL = DTLBUF;                         ! Point to buffer
5                 DTLTBL = 1                            ! Kill following check
5                 END
4             ;

4             IF  .DTLTBL NEQ 1           ! Unless we formatted it above
4                 AND
5                 (
5                 .DTLTBL EQL 0           ! If there is no detail table
5                 OR
6                 (
6                 IF  .DTLTBL NEQ 0       ! Interlock for not in table check
6                 THEN
6                 NOT NCP$TABLESEARCH (.DETAIL, .DTLTBL, DTL)
6                 ELSE
6                 TRUE                    ! Force conversion if not in table
6                 )
5                 )
```

```
  4             THEN
  5                 BEGIN                       ! Put out in some standard way
P 5                 $FAO
P 5                     (
P 5                     ASCID ('detail # !UW'),
P 5                     OUTLEN,
P 5                     UPLIT (DTLSIZ-1, DTLBUF+1),
P 5                     .DETAIL
  5                     );
  5                 DTLBUF [0] = .OUTLEN;       ! As counted string
  5                 DTL = DTLBUF
  5                 END
  4             END
  3         END
  2     ;
  2
  2     IF .CTR GEQU 4                              ! If there is enough for system
  2     THEN                                        ! Specific error text
  3         BEGIN
  4         IF .CTR GEQU (4 + .(.PTR+3) <0, 8, 0> )
  3         THEN                                    ! And the text is valid
  4             BEGIN
  4             ERR = .PTR + 3;                     ! Point to the counted string
  4             .LEN = .CTR - (.(.PTR+3) <0, 8, 0>) - 4; ! Adjust returned length
  5             .BFR = ..BFR + 4 + (.(.PTR+3) <0, 8, 0>) ! And buffer beyond it
  4             END
  3         ELSE                                    ! Tell the world its not clean
  3             ERR = UPLIT (%ASCIC '%NCP-W-ERRRSP, invalid error text in listener response')
  3         END
  2     ;
  2
  2
  2 !
  2 !       Signal the error to print it
  2 !
  2
  2     IF ..LEN NEQ 0                              ! Is there an entity for the message
  2         AND
  2         NOT .SHO                                ! and this is not a show or list
  2     THEN
  3         BEGIN
  3         .LEN = 0;                               ! Return no data to caller
  3         ENTDSC [0] = ENTSIZ - 1;                ! Descriptor for output is buffer
  3         ENTDSC [1] = ENTBUF + 1;                ! Less one byte for count
  3         ENT = ENTBUF;                           ! Set counted string address
  3         IF .NCP$GL_FNC_CODE NEQ NMA$C_FNC_TES ! Loop return with test data
  3         THEN
  4             BEGIN
  4             PTR = ..BFR;                        ! Set pointer to entity code
  4             NCP$FAOSET ();                      ! Setup conversion routines
  4             NCP$SHOENTITY (PTR);                ! Convert to fao parameters
  4             NCP$FAOL (ENTDSC);                  ! Convert to text
  4             ENTBUF [0] = .ENTDSC [0];           ! Make counted string
  4             END
  3         ELSE
  4             BEGIN
P 4             $FAO                                ! Convert test data if loop return
```

```
; 1569        P 1556  4                  (
; 1570        P 1557  4                  (
; 1571        P 1558  4                  IF .CODE EQL NMA$C_STS_PVA ! Special case the text for
; 1572        P 1559  4                  THEN ASCID ('Maximum data length = !UW') ! a loop message
; 1573        P 1560  4                  ELSE ASCID ('Messages not looped = !UW')
; 1574        P 1561  4                  ),
; 1575        P 1562  4                  OUTLEN,
; 1576        P 1563  4                  ENTDSC,                    ! Descriptor of buffer
; 1577        P 1564  4                  ...BFR                     ! Stack the data (word of loop count)
; 1578          1565  4                  );
; 1579          1566  4              ENTBUF [0] = .OUTLEN           ! Set counter for this message
; 1580          1567  4              END
; 1581          1568  3          END
; 1582          1569  2      ;
; 1583          1570  2
; 1584          1571  2      IF CH$RCHAR(.DTL) NEQ 0                ! If text following message,
; 1585          1572  2      THEN
; 1586          1573  2          COMMA = UPLIT(%ASCIC ',');         ! then delimit with a comma
; 1587          1574  2
; 1588          1575  2      IF
; 1589          1576  3          (
; 1590          1577  4          (
; 1591          1578  4          .CODE NEQ NMA$C_STS_MOR            ! If a not a success code
; 1592          1579  4          AND
; 1593          1580  4          .CODE NEQ NMA$C_STS_SUC
; 1594          1581  4          AND
; 1595          1582  4          .CODE NEQ NMA$C_STS_DON
; 1596          1583  4          AND
; 1597          1584  4          .CODE NEQ NMA$C_STS_PAR
; 1598          1585  4          )
; 1599          1586  3          AND
; 1600          1587  3          CH$RCHAR (.RSP) NEQ 0              ! and the response message is here
; 1601          1588  3          )
; 1602          1589  2          OR
; 1603          1590  2          CH$RCHAR (.DTL) NEQ 0              ! or any of the text strings are here
; 1604          1591  2          OR
; 1605          1592  2          CH$RCHAR (.ERR) NEQ 0              ! then print the error
; 1606          1593  2      THEN
; 1607          1594  2          SIGNAL (NCP$_NMLRSP, 5, .RSP, .COMMA, .DTL, .ENT, .ERR)
; 1608          1595  2      ;
; 1609          1596  2
; 1610          1597  2      RETURN .CODE                           ! Return data to caller
; 1611          1598  2
; 1612          1599  1      END;


                                                                     .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                          0005B           .BLKB   1
00 00 64 65 7A 69 6E 67 6F 63 65 72 6E 75 0C 0005C P.AAL:  .ASCII  <12>\unrecognized\<0><0><0>
                                             00 0006B
                                 00 00 00 00 0006C P.AAM:  .ASCII  <0><0><0><0>
                                 00 00 00 00 00070 P.AAN:  .ASCII  <0><0><0><0>
                                 00 00 00 00 00074 P.AAO:  .ASCII  <0><0><0><0>
                                 00 00 00 00 00078 P.AAP:  .ASCII  <0><0><0><0>
75 74 65 72 20 74 6E 65 6D 65 67 61 6E 61 6D 0007C P.AAR:  .ASCII  \management return # !SB\<0>
```

```
                  00  42  53  21  20  23  20  6E  72  0008B
                                           00000017  00094 P.AAQ:  .LONG    23                                        ;
                                           00000000' 00098         .ADDRESS P.AAR                                     ;
                                           0000001F  0009C P.AAS:  .LONG    31                                        ;
                                           00000000' 000A0         .ADDRESS RSPBUF+1                                  ;
                                           0000001F  000A4 P.AAT:  .LONG    31                                        ;
                                           00000000' 000A8         .ADDRESS DTLBUF+1                                  ;
          57  55  21  20  23  20  6C  69  61  74  65  64  000AC P.AAV:  .ASCII   \detail # !UW\                            ;
                                           0000000C  000B8 P.AAU:  .LONG    12                                        ;
                                           00000000' 000BC         .ADDRESS P.AAV                                     ;
                                           0000001F  000C0 P.AAW:  .LONG    31                                        ;
                                           00000000' 000C4         .ADDRESS DTLBUF+1                                  ;
2C  50  53  52  52  52  45  2D  57  2D  50  43  4E  25  36  000C8 P.AAX:  .ASCII   \6%NCP-W-ERRRSP, invalid error text in li\ ;
20  72  6F  72  72  65  20  64  69  6C  61  76  6E  69  20  000D7                                                    ;
                    69  6C  20  6E  69  20  74  78  65  74  000E6                                                    ;
65  73  6E  6F  70  73  65  72  20  72  65  6E  65  74  73  000F0         .ASCII   \stener response\<0>                     ;
                                                        00  000FF                                                    ;
65  6C  20  61  74  61  64  20  6D  75  6D  69  78  61  4D  00100 P.AAZ:  .ASCII   \Maximum data length = !UW\<0><0><0>      ;
        00  00  00  57  55  21  20  3D  20  68  74  67  6E  0010F                                                    ;
                                           00000019  0011C P.AAY:  .LONG    25                                        ;
                                           00000000' 00120         .ADDRESS P.AAZ                                     ;
6F  6C  20  74  6F  6E  20  73  65  67  61  73  73  65  4D  00124 P.ABB:  .ASCII   \Messages not looped = !UW\<0><0><0>      ;
        00  00  00  57  55  21  20  3D  20  64  65  70  6F  00133                                                    ;
                                           00000019  00140 P.ABA:  .LONG    25                                        ;
                                           00000000' 00144         .ADDRESS P.ABB                                     ;
                                    00  00  2C  01  00148 P.ABC:  .ASCII   <1>\,\<0><0>                              ;

                                                                  .PSECT   $OWN$,NOEXE,2

                                                     0007C DTLBUF: .BLKB    32
                                                     0009C RSPBUF: .BLKB    32
                                                     000BC ENTDSC: .BLKB    8
                                                     000C4 ENTBUF: .BLKB    32

                                                                  .EXTRN   NCP$_NMLRSP, NCP$GA_TBL_NMLSTS
                                                                  .EXTRN   NCP$GA_TBL_FOPDTL
                                                                  .EXTRN   NCP$GA_TBL_NCEDTL
                                                                  .EXTRN   NCP$GA_TBL_VMSENTDTL
                                                                  .EXTRN   NCP$GA_TBL_ENTDTL
                                                                  .EXTRN   NCP$GA_TBL_OPEDTL
                                                                  .EXTRN   NCP$FAOSET, NCP$SHOENTITY
                                                                  .EXTRN   NCP$FAOL, SYS$FAO

                                                                  .PSECT   $CODE$,NOWRT,2

                                                OFFC 00000        .ENTRY   NCP$READRSP, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 1158
                                                                           R10,R11                                    ;
                      5E          A4   AE  9E 00002               MOVAB    -92(SP), SP                                ;
                                  08   BC  D4 00006               CLRL     @LEN                                       ; 1265
                0C    BC 00000000'     00  9E 00009               MOVAB    NCP$GT_RSPBFR, @BFR                        ; 1266
                      52          04   AC  D0 00011               MOVL     LCB, R2                                    ; 1268
                      3A          0E   B2  91 00015               CMPB     @14(R2), #58                               ;
                                       47  12 00019               BNEQ     2$                                         ;
                      50 00000000'     00  9E 0001B               MOVAB    NML_RESP_QUEUE, R0                         ; 1275
                0C    AE          00   B0  0F 00022               REMQUE   @0(R0), ENTRY                              ;
                                       0F  1C 00027               BVC      1$                                         ;
                                       2C  DD 00029               PUSHL    #44                                        ; 1277
```

```
                         00000000G  8F   DD 0002B          PUSHL   #NCP$_NETIO                              ;
               00000000G  00        02   FB 00031          CALLS   #2, LIB$STOP                             ;
                          50    0C  AE   D0 00038 1$:      MOVL    ENTRY, R0                                ; 1278
                          57    08  A0   D0 0003C          MOVL    8(R0), CTR                               ;
                     24   AE 00000000'  00   9E 00040          MOVAB   NCP$GT_RSPBFR, PTR                   ; 1279
         24   BE     0C   A0        57   28 00048          MOVC3   CTR, 12(R0), @PTR                        ; 1280
                     10   AE    0C  A7   9E 0004E          MOVAB   12(R7), LENGTH                           ; 1281
                                0C  AE   9F 00053          PUSHAB  ENTRY                                    ; 1282
                                14  AE   9F 00056          PUSHAB  LENGTH                                   ;
               00000000G  00        02   FB 00059          CALLS   #2, LIB$FREE_VM                          ;
                                    3F   11 00060          BRB     3$                                       ; 1268
                                    7E   7C 00062 2$:      CLRQ    -(SP)                                    ; 1293
                                    7E   7C 00064          CLRQ    -(SP)                                    ;
                          7E  03E8  8F   3C 00066          MOVZWL  #1000, -(SP)                             ;
                         00000000'  00   9F 0006B          PUSHAB  NCP$GT_RSPBFR                            ;
                                    7E   7C 00071          CLRQ    -(SP)                                    ;
                                74  AE   9F 00073          PUSHAB  IOSB                                     ;
                                    31   DD 00076          PUSHL   #49                                      ;
                          7E    02  A2   3C 00078          MOVZWL  2(R2), -(SP)                             ;
                                    7E   D4 0007C          CLRL    -(SP)                                    ;
               00000000G  00        0C   FB 0007E          CALLS   #12, SYS$QIOW                            ;
                                54  AE   9F 00085          PUSHAB  IOSB                                     ; 1294
                                    50   DD 00088          PUSHL   STATUS                                   ;
                         00000000G  8F   DD 0008A          PUSHL   #NCP$_NETIO                              ;
                   FDE3   CF        03   FB 00090          CALLS   #3, NCP$SIGNETERR                        ;
                          57    56  AE   3C 00095          MOVZWL  IOSB+2, CTR                              ; 1296
                     24   AE 00000000'  00   9E 00099          MOVAB   NCP$GT_RSPBFR, PTR                   ; 1297
                     14   AE 00000000'  00   9E 000A1 3$:      MOVAB   P.AAL, RSP                           ; 1304
                     08   AE 00000000'  00   9E 000A9          MOVAB   P.AAM, COMMA                         ; 1305
                     20   AE 00000000'  00   9E 000B1          MOVAB   P.AAN, DTL                           ; 1306
                     04   AE 00000000'  00   9E 000B9          MOVAB   P.AAO, ENT                           ; 1307
                          6E 00000000'  00   9E 000C1          MOVAB   P.AAP, ERR                           ; 1308
                                    57   D5 000C8          TSTL    CTR                                      ; 1310
                                    1D   12 000CA          BNEQ    4$                                       ;
                                    6E   DD 000CC          PUSHL   ERR                                      ; 1312
                                08  AE   DD 000CE          PUSHL   ENT                                      ;
                                28  AE   DD 000D1          PUSHL   DTL                                      ;
                                14  AE   DD 000D4          PUSHL   COMMA                                    ;
                                24  AE   DD 000D7          PUSHL   RSP                                      ;
                                    05   DD 000DA          PUSHL   #5                                       ;
                         00000000G  8F   DD 000DC          PUSHL   #NCP$_NMLRSP                             ;
               00000000G  00        07   FB 000E2          CALLS   #7, LIB$STOP                             ;
                          58    24  AE   D0 000E9 4$:      MOVL    PTR, R8                                  ; 1315
                          5A        68   98 000ED          CVTBL   (R8), CODE                               ;
                                14  AE   9F 000F0          PUSHAB  RSP                                      ; 1318
                         00000000G  00   9F 000F3          PUSHAB  NCP$GA_TBL_NMLSTS                        ;
                          7E        5A   9A 000F9          MOVZBL  CODE, -(SP)                              ; 1319
               00000000V  00        03   FB 000FC          CALLS   #3, NCP$TABLESEARCH                      ;
                          28        50   E8 00103          BLBS    R0, 5$                                   ;
                                    5A   DD 00106          PUSHL   CODE                                     ; 1332
                         00000000'  00   9F 00108          PUSHAB  P.AAS                                    ;
                                30  AE   9F 0010E          PUSHAB  OUTLEN                                   ;
                         00000000'  00   9F 00111          PUSHAB  P.AAQ                                    ;
               00000000G  00        04   FB 00117          CALLS   #4, SYS$FAO                              ;
               00000000'  00    28  AE   90 0011E          MOVB    OUTLEN, RSPBUF                           ; 1333
                     14   AE 00000000'  00   9E 00126          MOVAB   RSPBUF, RSP                          ; 1334
                     1C   AE        01   CE 0012E 5$:      MNEGL   #1, DETAIL                               ; 1338
```

```
                                 03                57 D1 00132        CMPL     CTR, #3                                 ; 1340
                                                   03 18 00135        BGEQ     7$
                                                 01FD 31 00137 6$:    BRW      27$
                            1C   AE         01     A8 32 0013A 7$:    CVTWL    1(R8), DETAIL                           ; 1343
                      FFFFFFFF   8F         1C     AE D1 0013F        CMPL     DETAIL, #-1                             ; 1344
                                                   EE 13 00147        BEQL     6$
                      FFFFFFEE   8F                5A D1 00149        CMPL     CODE, #-18                              ; 1351
                                                   12 13 00150        BEQL     8$
                      FFFFFFF2   8F                5A D1 00152        CMPL     CODE, #-14
                                                   12 19 00159        BLSS     9$
                      FFFFFFF3   8F                5A D1 0015B        CMPL     CODE, #-13
                                                   09 14 00162        BGTR     9$
                                 59  00000000G     00 9E 00164 8$:    MOVAB    NCP$GA_TBL_FOPDTL, DTLTBL
                                                   6E 11 0016B        BRB      17$
                      FFFFFFEB   8F                5A D1 0016D 9$:    CMPL     CODE, #-21                              ; 1354
                                                   09 13 00174        BEQL     10$
                      FFFFFFED   8F                5A D1 00176        CMPL     CODE, #-19
                                                   09 12 0017D        BNEQ     11$
                                 59  00000000G     00 9E 0017F 10$:   MOVAB    NCP$GA_TBL_NCEDTL, DTLTBL
                                                   53 11 00186        BRB      17$
                      FFFFFFE7   8F                5A D1 00188 11$:   CMPL     CODE, #-25                              ; 1357
                                                   09 12 0018F        BNEQ     12$
                                 59  00000000G     00 9E 00191        MOVAB    NCP$GA_TBL_OPEDTL, DTLTBL
                                                   41 11 00198        BRB      17$
                      FFFFFFF5   8F                5A D1 0019A 12$:   CMPL     CODE, #-11                              ; 1360
                                                   12 13 001A1        BEQL     13$
                      FFFFFFF7   8F                5A D1 001A3        CMPL     CODE, #-9
                                                   23 19 001AA        BLSS     15$
                      FFFFFFF8   8F                5A D1 001AC        CMPL     CODE, #-8
                                                   1A 14 001B3        BGTR     15$
                                     00000000G     00 D5 001B5 13$:   TSTL     NCP$GL_ENTITY                           ; 1362
                                                   09 18 001BB        BGEQ     14$
                                 59  00000000G     00 9E 001BD        MOVAB    NCP$GA_TBL_VMSENTDTL, DTLTBL
                                                   15 11 001C4        BRB      17$
                                 59  00000000G     00 9E 001C6 14$:   MOVAB    NCP$GA_TBL_ENTDTL, DTLTBL
                                                   0C 11 001CD        BRB      17$
                                            1C     AE D5 001CF 15$:   TSTL     DETAIL                                  ; 1369
                                                   05 12 001D2        BNEQ     16$
                                 59                01 D0 001D4        MOVL     #1, DTLTBL
                                                   02 11 001D7        BRB      17$
                                                   59 D4 001D9 16$:   CLRL     DTLTBL
                      FFFFFFE7   8F                5A D1 001DB 17$:   CMPL     CODE, #-25                              ; 1378
                                                   0F 12 001E2        BNEQ     18$
                                 50  00000000G     00 D0 001E4        MOVL     NCP$GL_ENTITY, R0                       ; 1380
                                 01                50 D1 001EB        CMPL     R0, #1
                                                   08 13 001EE        BEQL     19$
                                 03                50 D1 001F0        CMPL     R0, #3                                  ; 1383
                                                   03 13 001F3 18$:   BEQL     19$
                                                 0093 31 001F5        BRW      23$
                                 53         2C     AE 9E 001F8 19$:   MOVAB    PREBUF, LOCPTR                          ; 1395
          04     A8         03   A8                2C 3A 001FC        LOCC     #44, 3(R8), 4(R8)                       ; 1407
                                                   02 12 00202        BNEQ     20$
                                                   51 D4 00204        CLRL     R1
                                 50         04     A8 9E 00206 20$:   MOVAB    4(R8), R0                               ; 1409
                 56              51                50 C3 0020A        SUBL3    R0, R1, PRELEN
                                 63       0A0D     8F B0 0020E        MOVW     #2573, (LOCPTR)                         ; 1411
56               20              6E                00 2C 00213        MOVC5    #0, (SP), #32, PRELEN, 2(LOCPTR)        ; 1412
```

```
                                02   A3     00218                                                   ;
                     63       202C   8F  B0 0021A         MOVW     #8236, (LOCPTR)                  ; 1413
                     56              04  C0 0021F         ADDL2    #4, PRELEN                       ; 1414
                     50         03   A8  9A 00222         MOVZBL   3(R8), R0                        ; 1418
                     53         04 A048  9E 00226         MOVAB    4(R0)[R8], LOCPTR                ; 1417
                                     5B  D4 0022B         CLRL     INDEX                            ; 1420
     37          1C  AE              5B  E1 0022D 21$:    BBC      INDEX, DETAIL, 22$               ; 1422
                                20   AE  9F 00232         PUSHAB   DTL                              ; 1424
                                     59  DD 00235         PUSHL    DTLTBL                           ;
                                     5B  DD 00237         PUSHL    INDEX                            ;
          00000000V  00              03  FB 00239         CALLS    #3, NCP$TABLESEARCH              ;
                     26              50  E9 00240         BLBC     R0, 22$                          ;
                     51         20   BE  9A 00243         MOVZBL   @DTL, R1                         ; 1426
                     51              56  C0 00247         ADDL2    PRELEN, R1                       ;
     50              58              53  C3 0024A         SUBL3    LOCPTR, R8, R0                   ; 1427
                     50       03E8   C0  9E 0024E         MOVAB    1000(R0), R0                     ;
                     50              51  D1 00253         CMPL     R1, R0                           ;
                                     11  14 00256         BGTR     22$                              ;
     63          2C  AE              56  28 00258         MOVC3    PRELEN, PREBUF, (LOCPTR)         ; 1434
                     50         20   AE  D0 0025D         MOVL     DTL, R0                          ; 1440
                     51              60  9A 00261         MOVZBL   (R0), R1                         ;
     63          01  A0              51  28 00264         MOVC3    R1, 1(R0), (LOCPTR)              ; 1442
     C0              5B              10  F3 00269 22$:    AOBLEQ   #16, INDEX, 21$                  ; 1420
                     53              58  C2 0026D         SUBL2    R8, R3                           ; 1448
03   A8              53              04  83 00270         SUBB3    #4, R3, 3(R8)                    ;
                     57              53  90 00275         MOVB     R3, CTR                          ; 1449
                     59              01  D0 00278         MOVL     #1, DTLTBL                       ; 1450
                         00000000'   00  94 0027B         CLRB     DTLBUF                           ; 1451
                 20  AE  00000000'   00  9E 00281         MOVAB    DTLBUF, DTL                      ; 1452
                                     68  11 00289         BRB      25$                              ; 1378
           FFFFFFF0  8F              5A  D1 0028B 23$:    CMPL     CODE, #-16                       ; 1457
                                     2D  13 00292         BEQL     24$                              ;
           FFFFFFE9  8F              5A  D1 00294         CMPL     CODE, #-23                       ; 1459
                                     24  13 0029B         BEQL     24$                              ;
           FFFFFFEA  8F              5A  D1 0029D         CMPL     CODE, #-22                       ; 1461
                                     1B  13 002A4         BEQL     24$                              ;
           FFFFFFFA  8F              5A  D1 002A6         CMPL     CODE, #-6                        ; 1463
                                     12  13 002AD         BEQL     24$                              ;
           FFFFFFE5  8F              5A  D1 002AF         CMPL     CODE, #-27                       ; 1465
                                     09  13 002B6         BEQL     24$                              ;
           FFFFFFE3  8F              5A  D1 002B8         CMPL     CODE, #-29                       ; 1467
                                     32  12 002BF         BNEQ     25$                              ;
                                18   AE  9F 002C1 24$:    PUSHAB   JUNK                             ; 1471
                                2C   AE  9F 002C4         PUSHAB   OUTLEN                           ;
                         00000000'   00  9F 002C7         PUSHAB   P.AAT                            ; 1476
                     7E              01  7D 002CD         MOVQ     #1, -(SP)                        ; 1471
                                30   AE  9F 002D0         PUSHAB   DETAIL                           ;
                         00000000G   00  DD 002D3         PUSHL    NCP$GL_ENTITY                    ; 1472
          00000000G  00              07  FB 002D9         CALLS    #7, NCP$FORMATPARM               ;
          00000000'  00         28   AE  90 002E0         MOVB     OUTLEN, DTLBUF                   ; 1480
                 20  AE  00000000'   00  9E 002E8         MOVAB    DTLBUF, DTL                      ; 1481
                     59              01  D0 002F0         MOVL     #1, DTLTBL                       ; 1482
                     01              59  D1 002F3 25$:    CMPL     DTLTBL, #1                       ; 1486
                                     3F  13 002F6         BEQL     27$                              ;
                                     59  D5 002F8         TSTL     DTLTBL                           ; 1489
                                     12  13 002FA         BEQL     26$                              ;
                                20   AE  9F 002FC         PUSHAB   DTL                              ; 1494
```

```
                          59   DD 002FF          PUSHL    DTLTBL                      ;
                      24  AE   DD 00301          PUSHL    DETAIL                      ;
00000000V   00            03   FB 00304          CALLS    #3, NCP$TABLESEARCH         ;
            29            50   E8 0030B          BLBS     R0, 27$                     ;
                      1C  AE   DD 0030E 26$:     PUSHL    DETAIL                      ; 1507
               00000000'  00   9F 00311          PUSHAB   P.AAW                       ;
                      30  AE   9F 00317          PUSHAB   OUTLEN                      ;
               00000000'  00   9F 0031A          PUSHAB   P.AAU                       ;
00000000G   00            04   FB 00320          CALLS    #4, SYS$FAO                 ;
00000000'   00        28  AE   90 00327          MOVB     OUTLEN, DTLBUF              ; 1508
       20   AE 00000000'  00   9E 0032F          MOVAB    DTLBUF, DTL                 ; 1509
            04            57   D1 00337 27$:     CMPL     CTR, #4                     ; 1515
                          32   1F 0033A          BLSSU    29$                         ;
            50        03  A8   9A 0033C          MOVZBL   3(R8), R0                   ; 1518
            50            04   C0 00340          ADDL2    #4, R0                      ;
            50            57   D1 00343          CMPL     CTR, R0                     ;
                          1F   1F 00346          BLSSU    28$                         ;
            6E        03  A8   9E 00348          MOVAB    3(R8), ERR                  ; 1521
            50        03  A8   9A 0034C          MOVZBL   3(R8), R0                   ; 1522
            57            50   C2 00350          SUBL2    R0, R7                      ;
       08   BC        FC  A7   9E 00353          MOVAB    -4(R7), @LEN                ;
            50        03  A8   9A 00358          MOVZBL   3(R8), R0                   ; 1523
            50        0C  BC   C0 0035C          ADDL2    @BFR, R0                    ;
       0C   BC        04  A0   9E 00360          MOVAB    4(R0), @BFR                 ;
                          07   11 00365          BRB      29$                         ;
            6E 00000000'  00   9E 00367 28$:     MOVAB    P.AAX, ERR                  ; 1526
                      08  BC   D5 0036E 29$:     TSTL     @LEN                        ; 1535
                          58   13 00371          BEQL     30$                         ;
            54        10  AC   E8 00373          BLBS     SHO, 30$                    ; 1537
                      08  BC   D4 00377          CLRL     @LEN                        ; 1540
00000000'   00            1F   D0 0037A          MOVL     #31, ENTDSC                 ; 1541
00000000'   00 00000000'  00   9E 00381          MOVAB    ENTBUF+1, ENTDSC+4          ; 1542
       04   AE 00000000'  00   9E 0038C          MOVAB    ENTBUF, ENT                 ; 1543
            12 00000000G  00   D1 00394          CMPL     NCP$GL_FNC_CODE, #18        ; 1544
                          30   13 0039B          BEQL     31$                         ;
       24   AE        0C  BC   D0 0039D          MOVL     @BFR, PTR                   ; 1547
00000000G   00            00   FB 003A2          CALLS    #0, NCP$FAOSET              ; 1548
                      24  AE   9F 003A9          PUSHAB   PTR                         ; 1549
00000000G   00            01   FB 003AC          CALLS    #1, NCP$SHOENTITY           ;
               00000000'  00   9F 003B3          PUSHAB   ENTDSC                      ; 1550
00000000G   00            01   FB 003B9          CALLS    #1, NCP$FAOL                ;
00000000'   00 00000000'  00   90 003C0          MOVB     ENTDSC, ENTBUF              ; 1551
                          39   11 003CB 30$:     BRB      34$                         ; 1544
            50        0C  BC   D0 003CD 31$:     MOVL     @BFR, R0                    ; 1565
                          60   DD 003D1          PUSHL    (R0)                        ;
               00000000'  00   9F 003D3          PUSHAB   ENTDSC                      ;
                      30  AE   9F 003D9          PUSHAB   OUTLEN                      ;
 FFFFFFF0   8F            5A   D1 003DC          CMPL     CODE, #-16                  ;
                          09   12 003E3          BNEQ     32$                         ;
            50 00000000'  00   9E 003E5          MOVAB    P.AAY, R0                   ;
                          07   11 003EC          BRB      33$                         ;
            50 00000000'  00   9E 003EE 32$:     MOVAB    P.ABA, R0                   ;
                          50   DD 003F5 33$:     PUSHL    R0                          ;
00000000G   00            04   FB 003F7          CALLS    #4, SYS$FAO                 ;
00000000'   00        28  AE   90 003FE          MOVB     OUTLEN, ENTBUF              ; 1566
                          50   D4 00406 34$:     CLRL     R0                          ; 1571
                      20  BE   95 00408          TSTB     @DTL                        ;
```

```
                                       0A  13 0040B         BEQL    35$                                        ;
                                       50  D6 0040D         INCL    R0                                         ;
                    08     AE 00000000' 00  9E 0040F         MOVAB   P.ABC, COMMA                               ; 1573
                           02           5A  D1 00417 35$:    CMPL    CODE, #2                                   ; 1578
                                       18  13 0041A         BEQL    36$                                        ;
                           01           5A  D1 0041C         CMPL    CODE, #1                                   ; 1580
                                       13  13 0041F         BEQL    36$                                        ;
              FFFFFF80     8F           5A  D1 00421         CMPL    CODE, #-128                                ; 1582
                                       0A  13 00428         BEQL    36$                                        ;
                           03           5A  D1 0042A         CMPL    CODE, #3                                   ; 1584
                                       05  13 0042D         BEQL    36$                                        ;
                                  14   BE  95 0042F         TSTB    @RSP                                       ; 1587
                                       08  12 00432         BNEQ    37$                                        ;
                           05           50  E8 00434 36$:    BLBS    R0, 37$                                    ; 1590
                                  00   BE  95 00437         TSTB    @ERR                                       ; 1592
                                       1D  13 0043A         BEQL    38$                                        ;
                                       6E  DD 0043C 37$:    PUSHL   ERR                                        ; 1594
                                  08   AE  DD 0043E         PUSHL   ENT                                        ;
                                  28   AE  DD 00441         PUSHL   DTL                                        ;
                                  14   AE  DD 00444         PUSHL   COMMA                                      ;
                                  24   AE  DD 00447         PUSHL   RSP                                        ;
                                       05  DD 0044A         PUSHL   #5                                         ;
                           00000000G   8F  DD 0044C         PUSHL   #NCP$_NMLRSP                               ;
              00000000G    00           07  FB 00452         CALLS   #7, LIB$SIGNAL                             ;
                           50           5A  D0 00459 38$:    MOVL    CODE, R0                                   ; 1597
                                           04 0045C         RET                                                ; 1599

; Routine Size:  1117 bytes,    Routine Base:  $CODE$ + 0580
```

```
1 %SBTTL  'NCP$CONERR  Decode an NML Response'
1 GLOBAL ROUTINE NCP$CONERR (COUNT, MSGBFR) =      !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is for the CONNECT routine to have an easy way
1 !       to process NICE error messages.
1 !       If the message is an error response, the error is signaled and
1 !       control does not return to the caller.
1 !       If the error status is SUC, DON or MOR, and there is a detail
1 !       or error message, an error is signaled to print these but
1 !       control returns normally to the caller.
1 !
1 !       If an error contains data, it is assumed to be an entity for the
1 !       error and the entity code is formatted and included in the error
1 !       message.  Entity codes may also occur with success codes and in
1 !       this case the data is printed as an entity if the message is not
1 !       a show or list command, indicated by the SHO parameter.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       COUNT   Length of buffer containing NICE message
1 !       MSGBFR  Address of buffer containing NICE message
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NCP$GL_ENTITY   Entity number sent in original message
1 !                       (If negative, then system-specific entity)
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Value of first byte of message, or error signalled
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2
2     LITERAL
2         RSPSIZ = 32,                    ! Size of response buffer required
2         DTLSIZ = 32,                    ! Size of detail buffer required
2         ENTSIZ = 32                     ! Size of entity code buffer
2         ;
2
2     LOCAL
2         STATUS,                         ! Service status return
2         OUTLEN,                         ! Length in a buffer
```

```
2           IOSB : BBLOCK [8],                  ! QIO status
2           CTR,                                ! General temps
2           PTR,
2           LEN,
2           BFR,
2           CODE,
2           ENTITY,                             ! Entity number (negative if sys-specific)
2           RSP,                                ! Pointer for response text
2           SHO,
2           COMMA,                              ! Pointer to separator before detail
2           DTL,                                ! Pointer for detail text
2           ERR,                                ! Pointer for error text
2           ENT,                                ! Pointer for entity code text
2           IDX,                                ! Index into tables
2           JUNK,                               ! Throw away temporary
2           DETAIL,                             ! Value of detail word
2           DTLTBL                              ! Address of detail table
2           ;
2
2       OWN
2           DTLBUF : VECTOR [DTLSIZ, BYTE],     ! Detail buffer
2           RSPBUF : VECTOR [RSPSIZ, BYTE],     ! Response buffer
2           ENTDSC : VECTOR [2],                ! Descriptor for string
2           ENTBUF : VECTOR [ENTSIZ, BYTE]      ! Entity string buffer
2           ;
2
2
2       EXTERNAL LITERAL
2           NCP$_NMLRSP,                        ! NML response message
2           NCP$_NETIO                          ! Network communication error
2           ;
2
2       EXTERNAL
2           NCP$GA_TBL_NMLSTS,                  ! NML status return codes
2           NCP$GA_TBL_FOPDTL,                  ! File operations detail codes
2           NCP$GA_TBL_NCEDTL,                  ! Network communications detail codes
2           NCP$GA_TBL_VMSENTDTL,               ! Detail table of VMS specific entities
2           NCP$GA_TBL_ENTDTL,                  ! Detail table of entities
2           NCP$GA_TBL_OPEDTL;                  ! Detail table of operation failures
2
2       EXTERNAL ROUTINE
2           NCP$FAOSET      : NOVALUE,          ! Setup to convert entity
2           NCP$SHOENTITY   : NOVALUE,          ! Convert entity
2           NCP$FAOL        : NOVALUE           ! Convert fao string for entity
2           ;
2
2
2       LEN = 0;                                ! Set callers data
2       BFR = NCP$GT_RSPBFR;
2
2           CH$MOVE (.COUNT, .MSGBFR, NCP$GT_RSPBFR);       ! Copy data into buffer
2
2           SHO = 0;
2           CTR = .COUNT;           ! Point and count into message
2           PTR = NCP$GT_RSPBFR;
2
2 !
```

```
  2 !     We need to set some defaults in case the message is bad
  2 !
  2
  2   RSP = UPLIT (%ASCIC 'unrecognized'); ! Some default text for message
  2   COMMA = UPLIT (%ASCIC '');
  2   DTL = UPLIT (%ASCIC '');
  2   ENT = UPLIT (%ASCIC '');
  2   ERR = UPLIT (%ASCIC '');
  2
  2   IF .CTR EQL 0                         ! If message is short, signal now
  2   THEN
  2       SIGNAL_STOP (NCP$_NMLRSP, 5, .RSP, .COMMA, .DTL, .ENT, .ERR)
  2   ;
  2
  2   CODE = .(.PTR) <0, 8, 1>;             ! First byte is a code
  2
  2   IF NOT NCP$TABLESEARCH                ! Find the code text if possible
  2           (
  2           .CODE <0, 8, 0>,              ! Code byte
  2           NCP$GA_TBL_NMLSTS,            ! Table
  2           RSP                           ! Return address of counted string
  2           )
  2
  2   THEN
  3       BEGIN
P 3       $FAO                              ! If not found, make some text
P 3           (
P 3           ASCID ('management return # !SB'),
P 3           OUTLEN,
P 3           UPLIT (RSPSIZ-1, RSPBUF+1),
P 3           .CODE
  3           );
  3       RSPBUF [0] = .OUTLEN;             ! As a counted string
  3       RSP = RSPBUF                      ! Point to it
  3       END
  2   ;
  2
  2   DETAIL = -1;                          ! No detail yet
  2
  2   IF .CTR GEQ 3                         ! Is there a detail word
  2   THEN
  3       BEGIN
  3       DETAIL = .(.PTR+1) <0, 16, 1>;    ! Obtain the word
  3       IF .DETAIL NEQ -1                 ! Ignore value?
  3       THEN
  4           BEGIN                         ! Nope
  4           DTLTBL =                      ! Find a table to use
  5               BEGIN
  5               SELECTONE .CODE OF
  5               SET
  5               [NMA$C_STS_FOP, NMA$C_STS_FIO, NMA$C_STS_FCO] :
  5                        NCP$GA_TBL_FOPDTL         ! File io errors
  5                        ;
  5               [NMA$C_STS_MLD, NMA$C_STS_MCF] :
  5                        NCP$GA_TBL_NCEDTL         ! Network io errors
  5                        ;
  5               [NMA$C_STS_OPE] :
```

```
5                         NCP$GA_TBL_OPEDTL          ! Operation failure
5                         ;
5                 [NMA$C_STS_CMP, NMA$C_STS_IDE, NMA$C_STS_STA] :
5                                                    ! Errors with entities
5                         IF .NCP$GL_ENTITY LSS 0 ! If system-specific entity
5                         THEN
5                                 NCP$GA_TBL_VMSENTDTL       ! VMS entities
5                         ELSE
5                                 NCP$GA_TBL_ENTDTL;         ! DNA entities
5                 [OTHERWISE] :                      ! Details not valid
6                         BEGIN
6                         IF .DETAIL EQL 0           ! Zero is null detail here
6                         THEN 1                     ! Null detail if not valid
6                         ELSE 0                     ! But report non zero detail
6                         END
5                         ;
5                 TES
5                 END
4             ;
4
4             IF  .CODE EQL NMA$C_STS_OPE            ! If operation failure
4             AND
5                 (.NCP$GL_ENTITY EQL                !  and entity is line
5                   NMA$C_ENT_LIN
5             OR
5                  .NCP$GL_ENTITY EQL                !  or circuit
5                   NMA$C_ENT_CIR)
5
4             THEN
5                 BEGIN
5                    LOCAL
5                    PREBUF : VECTOR [40, BYTE],     ! Buffer for string to proceed
5                                                    !  each detail message.
5                    PRELEN,                         ! Length of string to proceed
5                                                    !  each detail message.
5                    LOCPTR;                         ! Local pointer
5
5                 LOCPTR = PREBUF;                   ! Init pointer into buffer
5
5 !
5 ! Build the string which will preceed the detail text so that each detail
5 ! string output will line-up under the error text.  For example:
5 !
5 !       %facility-L-ident, error text            ! Original error message
5 !
5 !       %facility-L-ident, error text<CR><LF>    ! Message with two detail
5 !       <     SPACES      >, detail text<CR><LF> !  strings appended.
5 !       <     SPACES      >, detail text         !
5 !
6                 PRELEN = ( CH$FIND_CH(.(.PTR+3),! Get the number of characters
6                            .PTR + 4, %C',') )  !  in the facility and ident
5                               - (.PTR + 4);    !  portion of error message
5
5                 .LOCPTR <0, 16> = %X'0A0D';        ! Store <CR><LF> in buffer,
5                 LOCPTR = CH$FILL( %C' ', .PRELEN, .LOCPTR + 2 ); ! some spaces,
5                 .LOCPTR <0, 16> = %ASCII', ';      !  and a ", "
5                 PRELEN = .PRELEN + 4;              ! Length = length of facility
```

```
5                                           ! text plus <CR><LF> and ", "
5
5           LOCPTR = .PTR + 4 +             ! Point to end of original
5                     .(.PTR + 3) < 0, 8 >; !  error message text.
5
5           INCR INDEX FROM 0 TO 16 DO
6               BEGIN
6               IF .DETAIL < .INDEX, 1, 0 > ! If status or error bit is set,
6               AND                         !  and it's in the table,
6               NCP$TABLESEARCH (.INDEX, .DTLTBL, DTL)
6               AND                         !  and there's room in the
6               .PRELEN  + .(.DTL) < 0, 8 > !  response buffer.
6               LEQ .PTR + NCP$C_RSPSIZ - .LOCPTR
6               THEN
7                   BEGIN
7                   LOCPTR = CH$MOVE        ! Append the string which
7                               (           !  preceeds each detail message
7                               .PRELEN,    !  to the end of the error
7                               PREBUF,     !  message
7                               .LOCPTR     !
7                               );          !
7
7
7                   LOCPTR = CH$MOVE        ! Append detail to end of the
7                               (           !  error message
7                               .(.DTL) <0,8>,
7                               .DTL + 1,   !
7                               .LOCPTR     !
7                               );          !
6                   END;
5              END;
5
5           (.PTR + 3) < 0, 8 > =           ! Update message length.
5                     .LOCPTR - .PTR - 4;   !
5           CTR < 0, 8 > = .LOCPTR - .PTR;  ! Update counter.
5           DTLTBL = 1;                     ! Indicate that we formatted it
5           DTLBUF [0] = 0;                 ! Make sure we Don't print the
5           DTL = DTLBUF;                   !  detail #
5
5           END
5
4       ELSE
4       IF  .CODE EQL NMA$C_STS_PVA         ! Special details for these
4           OR                              ! Errors, its the parameter
4           .CODE EQL NMA$C_STS_PLO         ! name
4           OR
4           .CODE EQL NMA$C_STS_PNA
4           OR
4           .CODE EQL NMA$C_STS_PTY
4           OR
4           .CODE EQL NMA$C_STS_PGP
4           OR
4           .CODE EQL NMA$C_STS_PMS
4       THEN
5           BEGIN
5           NCP$FORMATPARM                  ! Format the parameter name
5               (
  5               .NCP$GL_ENTITY,                     ! Entity is here
  5               DETAIL,                             ! Parameter code is here
  5               TRUE,                               ! Give the name
  5               FALSE,                              ! Not the data
  5               UPLIT (DTLSIZ - 1, DTLBUF + 1), ! Describe the buffer
  5               OUTLEN,                             ! Length of text here
  5               JUNK                                ! Return pointer to throw away
  5               );
  5            DTLBUF [0] = .OUTLEN;                  ! Set length of counted string
  5            DTL = DTLBUF;                          ! Point to buffer
  5            DTLTBL = 1                             ! Kill following check
  5            END
  4         ;
  4
  4         IF   .DTLTBL NEQ 1           ! Unless we formatted it above
  4            AND
  5            (
  5            .DTLTBL EQL 0             ! If there is no detail table
  5            OR
  6            (
  6            IF   .DTLTBL NEQ 0        ! Interlock for not in table check
  6            THEN
  6            NOT NCP$TABLESEARCH (.DETAIL, .DTLTBL, DTL)
  6            ELSE
  6            TRUE                      ! Force conversion if not in table
  6            )
  5            )
  4         THEN
  5            BEGIN                     ! Put out in some standard way
P 5            $FAO
P 5               (
P 5               ASCID ('detail # !UW'),
P 5               OUTLEN,
P 5               UPLIT (DTLSIZ-1, DTLBUF+1),
P 5               .DETAIL
  5               );
  5            DTLBUF [0] = .OUTLEN;    ! As counted string
  5            DTL = DTLBUF
  5            END
  4         END
  3      END
  2   ;
  2
  2   IF .CTR GEQU 4                     ! If there is enough for system
  2   THEN                               ! Specific error text
  3      BEGIN
  4      IF .CTR GEQU (4 + .(.PTR+3) <0, 8, 0> )
  3      THEN                            ! And the text is valid
  4         BEGIN
  4         ERR = .PTR + 3;              ! Point to the counted string
  4         LEN = .CTR - (.(.PTR+3) <0, 8, 0>) - 4; ! Adjust returned length
  5         BFR = .BFR + 4 + (.(.PTR+3) <0, 8, 0>) ! And buffer beyond it
  4         END
  3      ELSE                            ! Tell the world its not clean
  3         ERR = UPLIT (%ASCIC '%NCP-W-ERRRSP, invalid error text in listener response')
  3      END
  2   ;
```

```
; 1956        1942  2
; 1957        1943  2
; 1958        1944  2  !
; 1959        1945  2  !      Signal the error to print it
; 1960        1946  2  !
; 1961        1947  2
; 1962        1948  2       IF .LEN NEQ 0                           ! Is there an entity for the message
; 1963        1949  2           AND
; 1964        1950  2           NOT .SHO                            ! and this is not a show or list
; 1965        1951  2       THEN
; 1966        1952  3           BEGIN
; 1967        1953  3           LEN = 0;                            ! Return no data to caller
; 1968        1954  3           ENTDSC [0] = ENTSIZ - 1;            ! Descriptor for output is buffer
; 1969        1955  3           ENTDSC [1] = ENTBUF + 1;            ! Less one byte for count
; 1970        1956  3           ENT = ENTBUF;                       ! Set counted string address
; 1971        1957  3           IF .NCP$GL_FNC_CODE NEQ NMA$C_FNC_TES ! Loop return with test data
; 1972        1958  3           THEN
; 1973        1959  4               BEGIN
; 1974        1960  4               PTR = .BFR;                     ! Set pointer to entity code
; 1975        1961  4               NCP$FAOSET ();                  ! Setup conversion routines
; 1976        1962  4               NCP$SHOENTITY (PTR);            ! Convert to fao parameters
; 1977        1963  4               NCP$FAOL (ENTDSC);              ! Convert to text
; 1978        1964  4               ENTBUF [0] = .ENTDSC [0];       ! Make counted string
; 1979        1965  4               END
; 1980        1966  3           ELSE
; 1981        1967  4               BEGIN
; 1982      P 1968  4               $FAO                            ! Convert test data if loop return
; 1983      P 1969  4                   (
; 1984      P 1970  4                   (
; 1985      P 1971  4                   IF .CODE EQL NMA$C_STS_PVA ! Special case the text for
; 1986      P 1972  4                   THEN ASCID ('Maximum data length = !UW') ! a loop message
; 1987      P 1973  4                   ELSE ASCID ('Messages not looped = !UW')
; 1988      P 1974  4                   ),
; 1989      P 1975  4                   OUTLEN,
; 1990      P 1976  4                   ENTDSC,                     ! Descriptor of buffer
; 1991      P 1977  4                   ..BFR                       ! Stack the data (word of loop count)
; 1992        1978  4                   );
; 1993        1979  4               ENTBUF [0] = .OUTLEN            ! Set counter for this message
; 1994        1980  4               END
; 1995        1981  3           END
; 1996        1982  2       ;
; 1997        1983  2
; 1998        1984  2       IF CH$RCHAR(.DTL) NEQ 0                 ! If text following message,
; 1999        1985  2       THEN
; 2000        1986  2           COMMA = UPLIT(%ASCIC ',');          ! then delimit with a comma
; 2001        1987  2
; 2002        1988  2       IF
; 2003        1989  3           (
; 2004        1990  4           (
; 2005        1991  4           .CODE NEQ NMA$C_STS_MOR             ! If a not a success code
; 2006        1992  4           AND
; 2007        1993  4           .CODE NEQ NMA$C_STS_SUC
; 2008        1994  4           AND
; 2009        1995  4           .CODE NEQ NMA$C_STS_DON
; 2010        1996  4           AND
; 2011        1997  4           .CODE NEQ NMA$C_STS_PAR
; 2012        1998  4           )
: 2013      1999 3           AND
: 2014      2000 3           CH$RCHAR (.RSP) NEQ 0          ! and the response message is here
: 2015      2001 3           )
: 2016      2002 2           OR
: 2017      2003 2           CH$RCHAR (.DTL) NEQ 0          ! or any of the text strings are here
: 2018      2004 2           OR
: 2019      2005 2           CH$RCHAR (.ERR) NEQ 0          ! then print the error
: 2020      2006 2       THEN
: 2021      2007 2           SIGNAL (NCP$_NMLRSP, 5, .RSP, .COMMA, .DTL, .ENT, .ERR)
: 2022      2008 2       ;
: 2023      2009 2
: 2024      2010 2       RETURN .CODE                       ! Return data to caller
: 2025      2011 2
: 2026      2012 1       END;
```

```
                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

00  00  64  65  7A  69  6E  67  6F  63  65  72  6E  75  0C  0014C P.ABD:  .ASCII  <12>\unrecognized\<0><0><0>
                                                        00  0015B
                                            00  00  00  00  0015C P.ABE:  .ASCII  <0><0><0><0>
                                            00  00  00  00  00160 P.ABF:  .ASCII  <0><0><0><0>
                                            00  00  00  00  00164 P.ABG:  .ASCII  <0><0><0><0>
                                            00  00  00  00  00168 P.ABH:  .ASCII  <0><0><0><0>
75  74  65  72  20  74  6E  65  6D  65  67  61  6E  61  6D  0016C P.ABJ:  .ASCII  \management return # !SB\<0>
                        00  42  53  21  20  23  20  6E  72  0017B
                                                  00000017  00184 P.ABI:  .LONG    23
                                                  00000000' 00188         .ADDRESS P.ABJ
                                                  0000001F  0018C P.ABK:  .LONG    31
                                                  00000000' 00190         .ADDRESS RSPBUF+1
                                                  0000001F  00194 P.ABL:  .LONG    31
                                                  00000000' 00198         .ADDRESS DTLBUF+1
            57  55  21  20  23  20  6C  69  61  74  65  64  0019C P.ABN:  .ASCII  \detail # !UW\
                                                  0000000C  001A8 P.ABM:  .LONG    12
                                                  00000000' 001AC         .ADDRESS P.ABN
                                                  0000001F  001B0 P.ABO:  .LONG    31
                                                  00000000' 001B4         .ADDRESS DTLBUF+1
2C  50  53  52  52  52  45  2D  57  2D  50  43  4E  25  36  001B8 P.ABP:  .ASCII  \6%NCP-W-ERRRSP, invalid error text in li\
20  72  6F  72  72  65  20  64  69  6C  61  76  6E  69  20  001C7
                    69  6C  20  6E  69  20  74  78  65  74  001D6
65  73  6E  6F  70  73  65  72  20  72  65  6E  65  74  73  001E0         .ASCII  \stener response\<0>
                                                        00  001EF
65  6C  20  61  74  61  64  20  6D  75  6D  69  78  61  4D  001F0 P.ABR:  .ASCII  \Maximum data length = !UW\<0><0><0>
        00  00  00  57  55  21  20  3D  20  68  74  67  6E  001FF
                                                  00000019  0020C P.ABQ:  .LONG    25
                                                  00000000' 00210         .ADDRESS P.ABR
6F  6C  20  74  6F  6E  20  73  65  67  61  73  73  65  4D  00214 P.ABT:  .ASCII  \Messages not looped = !UW\<0><0><0>
        00  00  00  57  55  21  20  3D  20  64  65  70  6F  00223
                                                  00000019  00230 P.ABS:  .LONG    25
                                                  00000000' 00234         .ADDRESS P.ABT
                                            00  00  2C  01  00238 P.ABU:  .ASCII  <1>\,\<0><0>

                                                                          .PSECT  $OWN$,NOEXE,2

                                                            000E4 DTLBUF: .BLKB    32
                                                            00104 RSPBUF: .BLKB    32
```

```
                                                              00124  ENTDSC: .BLKB   8
                                                              0012C  ENTBUF: .BLKB   32


                                                                             .PSECT  $CODE$,NOWRT,2

                                                         0FFC 00000          .ENTRY  NCP$CONERR, Save R2,R3,R4,R5,R6,R7,R8,R9,-    ; 1601
                                                                                     R10,R11                                       ;
                                         5E        A0  AE  9E 00002          MOVAB   -96(SP), SP                                   ;
                                                       58  D4 00006          CLRL    LEN                                           ; 1704
                                         5B 00000000'  00  9E 00008          MOVAB   NCP$GT_RSPBFR, BFR                            ; 1705
               00000000'  00        08   BC        04  AC  28 0000F          MOVC3   COUNT, @MSGBFR, NCP$GT_RSPBFR                 ; 1707
                                                   0C  AE  D4 00019          CLRL    SHO                                           ; 1709
                                         57        04  AC  D0 0001C          MOVL    COUNT, CTR                                    ; 1710
                                    28   AE 00000000'  00  9E 00020          MOVAB   NCP$GT_RSPBFR, PTR                            ; 1711
                                    18   AE 00000000'  00  9E 00028          MOVAB   P.ABD, RSP                                    ; 1717
                                    08   AE 00000000'  00  9E 00030          MOVAB   P.ABE, COMMA                                  ; 1718
                                    24   AE 00000000'  00  9E 00038          MOVAB   P.ABF, DTL                                    ; 1719
                                    04   AE 00000000'  00  9E 00040          MOVAB   P.ABG, ENT                                    ; 1720
                                         6E 00000000'  00  9E 00048          MOVAB   P.ABH, ERR                                    ; 1721
                                                       57  D5 0004F          TSTL    CTR                                           ; 1723
                                                       1D  12 00051          BNEQ    1$                                            ;
                                                       6E  DD 00053          PUSHL   ERR                                           ; 1725
                                                   08  AE  DD 00055          PUSHL   ENT                                           ;
                                                   2C  AE  DD 00058          PUSHL   DTL                                           ;
                                                   14  AE  DD 0005B          PUSHL   COMMA                                         ;
                                                   28  AE  DD 0005E          PUSHL   RSP                                           ;
                                                       05  DD 00061          PUSHL   #5                                            ;
                                            00000000G  8F  DD 00063          PUSHL   #NCP$_NMLRSP                                  ;
                             00000000G   00            07  FB 00069          CALLS   #7, LIB$STOP                                  ;
                                         59        28  AE  D0 00070  1$:     MOVL    PTR, R9                                       ; 1728
                                    10   AE            69  98 00074          CVTBL   (R9), CODE                                    ;
                                                   18  AE  9F 00078          PUSHAB  RSP                                           ; 1731
                                            00000000G  00  9F 0007B          PUSHAB  NCP$GA_TBL_NMLSTS                             ;
                                         7E        18  AE  9A 00081          MOVZBL  CODE, -(SP)                                   ; 1732
                             00000000V   00            03  FB 00085          CALLS   #3, NCP$TABLESEARCH                           ;
                                         29            50  E8 0008C          BLBS    R0, 2$                                        ;
                                                   10  AE  DD 0008F          PUSHL   CODE                                          ; 1745
                                            00000000'  00  9F 00092          PUSHAB  P.ABK                                         ;
                                                   34  AE  9F 00098          PUSHAB  OUTLEN                                        ;
                                            00000000'  00  9F 0009B          PUSHAB  P.ABI                                         ;
                             00000000G   00            04  FB 000A1          CALLS   #4, SYS$FAO                                   ;
                             00000000'   00        2C  AE  90 000A8          MOVB    OUTLEN, RSPBUF                                ; 1746
                                    18   AE 00000000'  00  9E 000B0          MOVAB   RSPBUF, RSP                                   ; 1747
                                    20   AE            01  CE 000B8  2$:     MNEGL   #1, DETAIL                                    ; 1751
                                         03            57  D1 000BC          CMPL    CTR, #3                                       ; 1753
                                                       03  18 000BF          BGEQ    4$                                            ;
                                                     0217  31 000C1  3$:     BRW     24$                                           ;
                                    20   AE        01  A9  32 000C4  4$:     CVTWL   1(R9), DETAIL                                 ; 1756
                              FFFFFFFF   8F        20  AE  D1 000C9          CMPL    DETAIL, #-1                                   ; 1757
                                                       EE  13 000D1          BEQL    3$                                            ;
                              FFFFFFEE   8F        10  AE  D1 000D3          CMPL    CODE, #-18                                    ; 1764
                                                       14  13 000DB          BEQL    5$                                            ;
                              FFFFFFF2   8F        10  AE  D1 000DD          CMPL    CODE, #-14                                    ;
                                                       13  19 000E5          BLSS    6$                                            ;
                              FFFFFFF3   8F        10  AE  D1 000E7          CMPL    CODE, #-13                                    ;
```

```
                                        09  14 000EF         BGTR    6$                                        :
                        5A 00000000G    00  9E 000F1  5$:    MOVAB   NCP$GA_TBL_FOPDTL, DTLTBL                 :
                                        74  11 000F8         BRB     14$                                       :
               FFFFFFEB 8F         10   AE  D1 000FA  6$:    CMPL    CODE, #-21                                : 1767
                                        0A  13 00102         BEQL    7$                                        :
               FFFFFFED 8F         10   AE  D1 00104         CMPL    CODE, #-19                                :
                                        09  12 0010C         BNEQ    8$                                        :
                        5A 00000000G    00  9E 0010E  7$:    MOVAB   NCP$GA_TBL_NCEDTL, DTLTBL                 :
                                        57  11 00115         BRB     14$                                       :
               FFFFFFE7 8F         10   AE  D1 00117  8$:    CMPL    CODE, #-25                                : 1770
                                        09  12 0011F         BNEQ    9$                                        :
                        5A 00000000G    00  9E 00121         MOVAB   NCP$GA_TBL_OPEDTL, DTLTBL                 :
                                        44  11 00128         BRB     14$                                       :
               FFFFFFF5 8F         10   AE  D1 0012A  9$:    CMPL    CODE, #-11                                : 1773
                                        14  13 00132         BEQL    10$                                       :
               FFFFFFF7 8F         10   AE  D1 00134         CMPL    CODE, #-9                                 :
                                        24  19 0013C         BLSS    12$                                       :
               FFFFFFF8 8F         10   AE  D1 0013E         CMPL    CODE, #-8                                 :
                                        1A  14 00146         BGTR    12$                                       :
                           00000000G    00  D5 00148 10$:    TSTL    NCP$GL_ENTITY                             : 1775
                                        09  18 0014E         BGEQ    11$                                       :
                        5A 00000000G    00  9E 00150         MOVAB   NCP$GA_TBL_VMSENTDTL, DTLTBL              :
                                        15  11 00157         BRB     14$                                       :
                        5A 00000000G    00  9E 00159 11$:    MOVAB   NCP$GA_TBL_ENTDTL, DTLTBL                 :
                                        0C  11 00160         BRB     14$                                       :
                                   20   AE  D5 00162 12$:    TSTL    DETAIL                                    : 1782
                                        05  12 00165         BNEQ    13$                                       :
                        5A              01  D0 00167         MOVL    #1, DTLTBL                                :
                                        02  11 0016A         BRB     14$                                       :
                                        5A  D4 0016C 13$:    CLRL    DTLTBL                                    :
               FFFFFFE7 8F         10   AE  D1 0016E 14$:    CMPL    CODE, #-25                                : 1791
                                        0F  12 00176         BNEQ    15$                                       :
                        50 00000000G    00  D0 00178         MOVL    NCP$GL_ENTITY, R0                         : 1793
                        01              50  D1 0017F         CMPL    R0, #1                                    :
                                        08  13 00182         BEQL    16$                                       :
                        03              50  D1 00184         CMPL    R0, #3                                    : 1796
                                        03  13 00187 15$:    BEQL    16$                                       :
                                      009D  31 00189         BRW     20$                                       :
                        53         30   AE  9E 0018C 16$:    MOVAB   PREBUF, LOCPTR                            : 1808
          04      A9       03       A9  2C  3A 00190         LOCC    #44, 3(R9), 4(R9)                         : 1820
                                        02  12 00196         BNEQ    17$                                       :
                                        51  D4 00198         CLRL    R1                                        :
                        50         04   A9  9E 0019A 17$:    MOVAB   4(R9), R0                                 : 1822
                  56    51              50  C3 0019E         SUBL3   R0, R1, PRELEN                            :
                        50              53  3C 001A2         MOVZWL  LOCPTR, R0                                : 1824
                        60       0A0D   8F  3C 001A5         MOVZWL  #2573, (R0)                               :
     56           20    6E              00  2C 001AA         MOVC5   #0, (SP), #32, PRELEN, 2(LOCPTR)          : 1825
                                   02   A3     001AF                                                           :
                        50              53  3C 001B1         MOVZWL  LOCPTR, R0                                : 1826
                        60       202C   8F  3C 001B4         MOVZWL  #8236, (R0)                               :
                        56              04  C0 001B9         ADDL2   #4, PRELEN                                : 1827
                        50         03   A9  9A 001BC         MOVZBL  3(R9), R0                                 : 1831
                        53       04 A049    9E 001C0         MOVAB   4(R0)[R9], LOCPTR                         : 1830
                                   14   AE  D4 001C5         CLRL    INDEX                                     : 1833
                  38       20   AE 14   AE  E1 001C8 18$:    BBC     INDEX, DETAIL, 19$                        : 1835
                                   24   AE  9F 001CE         PUSHAB  DTL                                       : 1837
                                        5A  DD 001D1         PUSHL   DTLTBL                                    :
```

```
                                 1C   AE   DD 001D3         PUSHL   INDEX                                   :
                  00000000V  00       03   FB 001D6         CALLS   #3, NCP$TABLESEARCH                     :
                             26       50   E9 001DD         BLBC    R0, 19$                                 :
                             51  24   BE   9A 001E0         MOVZBL  @DTL, R1                                : 1839
                             51       56   C0 001E4         ADDL2   PRELEN, R1                              :
           50                59       53   C3 001E7         SUBL3   LOCPTR, R9, R0                          : 1840
                             50 03E8  C0   9E 001EB         MOVAB   1000(R0), R0                            :
                             50       51   D1 001F0         CMPL    R1, R0                                  :
                                      11   14 001F3         BGTR    19$                                     :
           63         30     AE       56   28 001F5         MOVC3   PRELEN, PREBUF, (LOCPTR)                : 1847
                             50  24   AE   D0 001FA         MOVL    DTL, R0                                 : 1853
                             51       60   9A 001FE         MOVZBL  (R0), R1                                :
           63         01     A0       51   28 00201         MOVC3   R1, 1(R0), (LOCPTR)                     : 1855
           BD         14     AE       10   F3 00206  19$:   AOBLEQ  #16, INDEX, 18$                         : 1833
                             53       59   C2 0020B         SUBL2   R9, R3                                  : 1861
      03   A9                53       04   83 0020E         SUBB3   #4, R3, 3(R9)                           :
                             57       53   90 00213         MOVB    R3, CTR                                 : 1862
                             5A       01   D0 00216         MOVL    #1, DTLTBL                              : 1863
                                00000000'  00   94 00219    CLRB    DTLBUF                                  : 1864
                      24     AE 00000000'  00   9E 0021F    MOVAB   DTLBUF, DTL                             : 1865
                                      6E   11 00227         BRB     22$                                     : 1791
                FFFFFFF0     8F  10   AE   D1 00229  20$:   CMPL    CODE, #-16                              : 1870
                                      32   13 00231         BEQL    21$                                     :
                FFFFFFE9     8F  10   AE   D1 00233         CMPL    CODE, #-23                              : 1872
                                      28   13 0023B         BEQL    21$                                     :
                FFFFFFEA     8F  10   AE   D1 0023D         CMPL    CODE, #-22                              : 1874
                                      1E   13 00245         BEQL    21$                                     :
                FFFFFFFA     8F  10   AE   D1 00247         CMPL    CODE, #-6                               : 1876
                                      14   13 0024F         BEQL    21$                                     :
                FFFFFFE5     8F  10   AE   D1 00251         CMPL    CODE, #-27                              : 1878
                                      0A   13 00259         BEQL    21$                                     :
                FFFFFFE3     8F  10   AE   D1 0025B         CMPL    CODE, #-29                              : 1880
                                      32   12 00263         BNEQ    22$                                     :
                                 1C   AE   9F 00265  21$:   PUSHAB  JUNK                                    : 1884
                                 30   AE   9F 00268         PUSHAB  OUTLEN                                  :
                                00000000'  00   9F 0026B    PUSHAB  P.ABL                                   : 1889
                             7E       01   7D 00271         MOVQ    #1, -(SP)                               : 1884
                                 34   AE   9F 00274         PUSHAB  DETAIL                                  :
                                00000000G  00   DD 00277    PUSHL   NCP$GL ENTITY                           : 1885
                00000000G    00       07   FB 0027D         CALLS   #7, NCP$FORMATPARM                      :
                00000000'    00  2C   AE   90 00284         MOVB    OUTLEN, DTLBUF                          : 1893
                      24     AE 00000000'  00   9E 0028C    MOVAB   DTLBUF, DTL                             : 1894
                             5A       01   D0 00294         MOVL    #1, DTLTBL                              : 1895
                             01       5A   D1 00297  22$:   CMPL    DTLTBL, #1                              : 1899
                                      3F   13 0029A         BEQL    24$                                     :
                                      5A   D5 0029C         TSTL    DTLTBL                                  : 1902
                                      12   13 0029E         BEQL    23$                                     :
                                 24   AE   9F 002A0         PUSHAB  DTL                                     : 1907
                                      5A   DD 002A3         PUSHL   DTLTBL                                  :
                                 28   AE   DD 002A5         PUSHL   DETAIL                                  :
                00000000V    00       03   FB 002A8         CALLS   #3, NCP$TABLESEARCH                     :
                             29       50   E8 002AF         BLBS    R0, 24$                                 :
                                 20   AE   DD 002B2  23$:   PUSHL   DETAIL                                  : 1920
                                00000000'  00   9F 002B5    PUSHAB  P.ABO                                   :
                                 34   AE   9F 002BB         PUSHAB  OUTLEN                                  :
                                00000000'  00   9F 002BE    PUSHAB  P.ABM                                   :
                00000000G    00       04   FB 002C4         CALLS   #4, SYS$FAO                             :
```

```
00000000'  00        2C  AE  90 002CB           MOVB     OUTLEN, DTLBUF                ; 1921
       24  AE 00000000'  00  9E 002D3           MOVAB    DTLBUF, DTL                   ; 1922
           04            57  D1 002DB 24$:      CMPL     CTR, #4                       ; 1928
                         26  1F 002DE           BLSSU    26$                           ;
           50        03  A9  9A 002E0           MOVZBL   3(R9), R0                     ; 1931
           51        04  A0  9E 002E4           MOVAB    4(R0), R1                     ;
           51            57  D1 002E8           CMPL     CTR, R1                       ;
                         12  1F 002EB           BLSSU    25$                           ;
           6E        03  A9  9E 002ED           MOVAB    3(R9), ERR                    ; 1934
           57            50  C2 002F1           SUBL2    R0, R7                        ; 1935
           58        FC  A7  9E 002F4           MOVAB    -4(R7), LEN                   ;
           5B        04 A04B 9E 002F8           MOVAB    4(R0)[BFR], BFR               ; 1936
                         07  11 002FD           BRB      26$                           ;
           6E 00000000'  00  9E 002FF 25$:      MOVAB    P.ABP, ERR                    ; 1939
                         58  D5 00306 26$:      TSTL     LEN                           ; 1948
                         56  13 00308           BEQL     27$                           ;
           52        0C  AE  E8 0030A           BLBS     SHO, 27$                      ; 1950
                         58  D4 0030E           CLRL     LEN                           ; 1953
00000000'  00            1F  D0 00310           MOVL     #31, ENTDSC                   ; 1954
00000000'  00 00000000'  00  9E 00317           MOVAB    ENTBUF+1, ENTDSC+4            ; 1955
       04  AE 00000000'  00  9E 00322           MOVAB    ENTBUF, ENT                   ; 1956
           12 00000000G  00  D1 0032A           CMPL     NCP$GL_FNC_CODE, #18          ; 1957
                         2F  13 00331           BEQL     28$                           ;
       28  AE            5B  D0 00333           MOVL     BFR, PTR                      ; 1960
00000000G  00            00  FB 00337           CALLS    #0, NCP$FAOSET                ; 1961
                     28  AE  9F 0033E           PUSHAB   PTR                           ; 1962
00000000G  00            01  FB 00341           CALLS    #1, NCP$SHOENTITY             ;
              00000000'  00  9F 00348           PUSHAB   ENTDSC                        ; 1963
00000000G  00            01  FB 0034E           CALLS    #1, NCP$FAOL                  ;
00000000'  00 00000000'  00  90 00355           MOVB     ENTDSC, ENTBUF                ; 1964
                         36  11 00360 27$:      BRB      31$                           ; 1957
                         6B  DD 00362 28$:      PUSHL    (BFR)                         ; 1978
              00000000'  00  9F 00364           PUSHAB   ENTDSC                        ;
                     34  AE  9F 0036A           PUSHAB   OUTLEN                        ;
FFFFFFF0   8F        1C  AE  D1 0036D           CMPL     CODE, #-16                    ;
                         09  12 00375           BNEQ     29$                           ;
           50 00000000'  00  9E 00377           MOVAB    P.ABQ, R0                     ;
                         07  11 0037E           BRB      30$                           ;
           50 00000000'  00  9E 00380 29$:      MOVAB    P.ABS, R0                     ;
                         50  DD 00387 30$:      PUSHL    R0                            ;
00000000G  00            04  FB 00389           CALLS    #4, SYS$FAO                   ;
00000000'  00        2C  AE  90 00390           MOVB     OUTLEN, ENTBUF                ; 1979
                         50  D4 00398 31$:      CLRL     R0                            ; 1984
                     24  BE  95 0039A           TSTB     @DTL                          ;
                         0A  13 0039D           BEQL     32$                           ;
                         50  D6 0039F           INCL     R0                            ;
       08  AE 00000000'  00  9E 003A1           MOVAB    P.ABU, COMMA                  ; 1986
           02        10  AE  D1 003A9 32$:      CMPL     CODE, #2                      ; 1991
                         1B  13 003AD           BEQL     33$                           ;
           01        10  AE  D1 003AF           CMPL     CODE, #1                      ; 1993
                         15  13 003B3           BEQL     33$                           ;
FFFFFF80   8F        10  AE  D1 003B5           CMPL     CODE, #-128                   ; 1995
                         0B  13 003BD           BEQL     33$                           ;
           03        10  AE  D1 003BF           CMPL     CODE, #3                      ; 1997
                         05  13 003C3           BEQL     33$                           ;
                     18  BE  95 003C5           TSTB     @RSP                          ; 2000
                         08  12 003C8           BNEQ     34$                           ;
```

```
                              05           50  E8 003CA 33$:     BLBS    R0, 34$                                  ; 2003
                                    00     BE  95 003CD          TSTB    @ERR                                     ; 2005
                                           1D  13 003D0          BEQL    35$                                      ;
                                           6E  DD 003D2 34$:     PUSHL   ERR                                      ; 2007
                                    08     AE  DD 003D4          PUSHL   ENT                                      ;
                                    2C     AE  DD 003D7          PUSHL   DTL                                      ;
                                    14     AE  DD 003DA          PUSHL   COMMA                                    ;
                                    28     AE  DD 003DD          PUSHL   RSP                                      ;
                                           05  DD 003E0          PUSHL   #5                                       ;
                              00000000G    8F  DD 003E2          PUSHL   #NCP$_NMLRSP                             ;
                   00000000G  00           07  FB 003E8          CALLS   #7, LIB$SIGNAL                           ;
                              50    10     AE  D0 003EF 35$:     MOVL    CODE, R0                                 ; 2010
                                               04 003F3          RET                                              ; 2012

; Routine Size:  1012 bytes,    Routine Base:  $CODE$ + 09DD
```

```
1 %SBTTL  'NCP$TABLESEARCH  Find an Entry in a Text Table'
1 GLOBAL ROUTINE NCP$TABLESEARCH (CODE, TBL, RTXTC) =     !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine searches a table for a word code and returns an
1 !       address of a counted string of an associated text string.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       CODE            Value of the code word
1 !       TBL             Address of the table
1 !       RTXTC           Address to return the address of the counted string
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Success or failure  RTXTC set to 'unrecognized' if failure
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     LOCAL                                 ! Pointer to the table
2         TPTR : REF BBLOCKVECTOR [1, 4]
2         ;               !
2
2     .RTXTC = UPLIT (%ASCIC 'unrecognized');
2     TPTR = .TBL;
2
2     INCRU IDX FROM 0                      ! Scan the table
2     DO
3         BEGIN
3         IF .TPTR [.IDX, 0, 0, 16, 1]      ! Look for the end first
3               EQL                         ! Use a signed reference for this
3               -1
3         THEN
3             RETURN FAILURE                ! Not found, return failure
3         ;
3
3         IF .TPTR [.IDX, 0, 0, 16, 0]      ! Look for the code (unsigned)
3               EQL
3           .CODE <0, 16, 0>                ! Code as a word
3         THEN
```

```
; 2085          2070  4                 BEGIN                       ! Return the real address
; 2086          2071  4                 .RTXTC = .TPTR [.IDX, 2, 0, 16, 1] ! Make address from the offset
; 2087          2072  4                     + TPTR [.IDX, 2, 0, 16, 1];
; 2088          2073  4                 RETURN SUCCESS              ! We found it
; 2089          2074  4                 END
; 2090          2075  3             END
; 2091          2076  2         ;
; 2092          2077  2         RETURN FAILURE                      ! Better never fail this way
; 2093          2078  2
; 2094          2079  1         END;


                                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

00  00  64  65  7A  69  6E  67  6F  63  65  72  6E  75  0C  0023C P.ABV:  .ASCII  <12>\unrecognized\<0><0><0>                      ;
                                                        00  0024B                                                                  ;


                                                                  .PSECT  $CODE$,NOWRT,2

                                               000C 00000         .ENTRY  NCP$TABLESEARCH, Save R2,R3                         ; 2014
                       0C    BC 00000000'  00    9E 00002         MOVAB   P.ABV, @RTXTC                                       ; 2053
                             52        08  AC    D0 0000A         MOVL    TBL, TPTR                                           ; 2054
                                           51    D4 0000E         CLRL    IDX                                                 ; 2068
                             50          6241    DE 00010 1$:     MOVAL   (TPTR)[IDX], R0                                     ; 2059
                     FFFF    8F            60    B1 00014         CMPW    (R0), #-1                                           ; 2061
                                           19    13 00019         BEQL    3$                                                  ;
                       04    AC            60    B1 0001B         CMPW    (R0), CODE                                          ; 2068
                                           0F    12 0001F         BNEQ    2$                                                  ;
                             50            02    C0 00021         ADDL2   #2, R0                                              ; 2071
                             53            60    32 00024         CVTWL   (R0), R3                                            ; 2072
             0C    BC        53            50    C1 00027         ADDL3   R0, R3, @RTXTC                                      ;
                             50            01    D0 0002C         MOVL    #1, R0                                              ; 2073
                                                 04 0002F         RET                                                         ;
                                           51    D6 00030 2$:     INCL    IDX                                                 ; 2066
                                           DC    11 00032         BRB     1$                                                  ;
                                           50    D4 00034 3$:     CLRL    R0                                                  ; 2079
                                                 04 00036         RET                                                         ;

; Routine Size:  55 bytes,    Routine Base:  $CODE$ + 0DD1
; 2096          2080  1 END                                       !End of module
; 2097          2081  0 ELUDOM



                                                                          .EXTRN  LIB$SIGNAL, LIB$STOP

;                                     PSECT SUMMARY
;
;       Name                        Bytes                         Attributes
;
;   $PLIT$                              588  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $GLOBAL$                           1288  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                               332  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                             3592  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                              Library Statistics
;
;                                             -------- Symbols --------        Pages        Processing
;       File                                  Total    Loaded   Percent        Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1          9776        22         0          581         00:01.0
;   _$255$DUA28:[NCP.OBJ]NMALIBRY.L32;1         887        22         2           47         00:00.7
;   _$255$DUA28:[NCP.OBJ]NCPLIBRY.L32;1         373        22         5           52         00:00.3





;                                     COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:NCPNETIO/OBJ=OBJ$:NCPNETIO MSRC$:NCPNETIO/UPDATE=(ENH$:NCPNETIO)

; Size:           3592 code + 2208 data bytes
; Run Time:          00:54.9
; Elapsed Time:      02:37.2
; Lines/CPU Min:      2273
; Lexemes/CPU-Min: 15466
; Memory Used:  267 pages
; Compilation Complete
```

0267 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NCPCONCAR
LIS

NCPERRMSG
LIS

NCPCONMAN
LIS

NCPLIBRY
B32

NCPMAIN
LIS

NCPNETIO
LIS

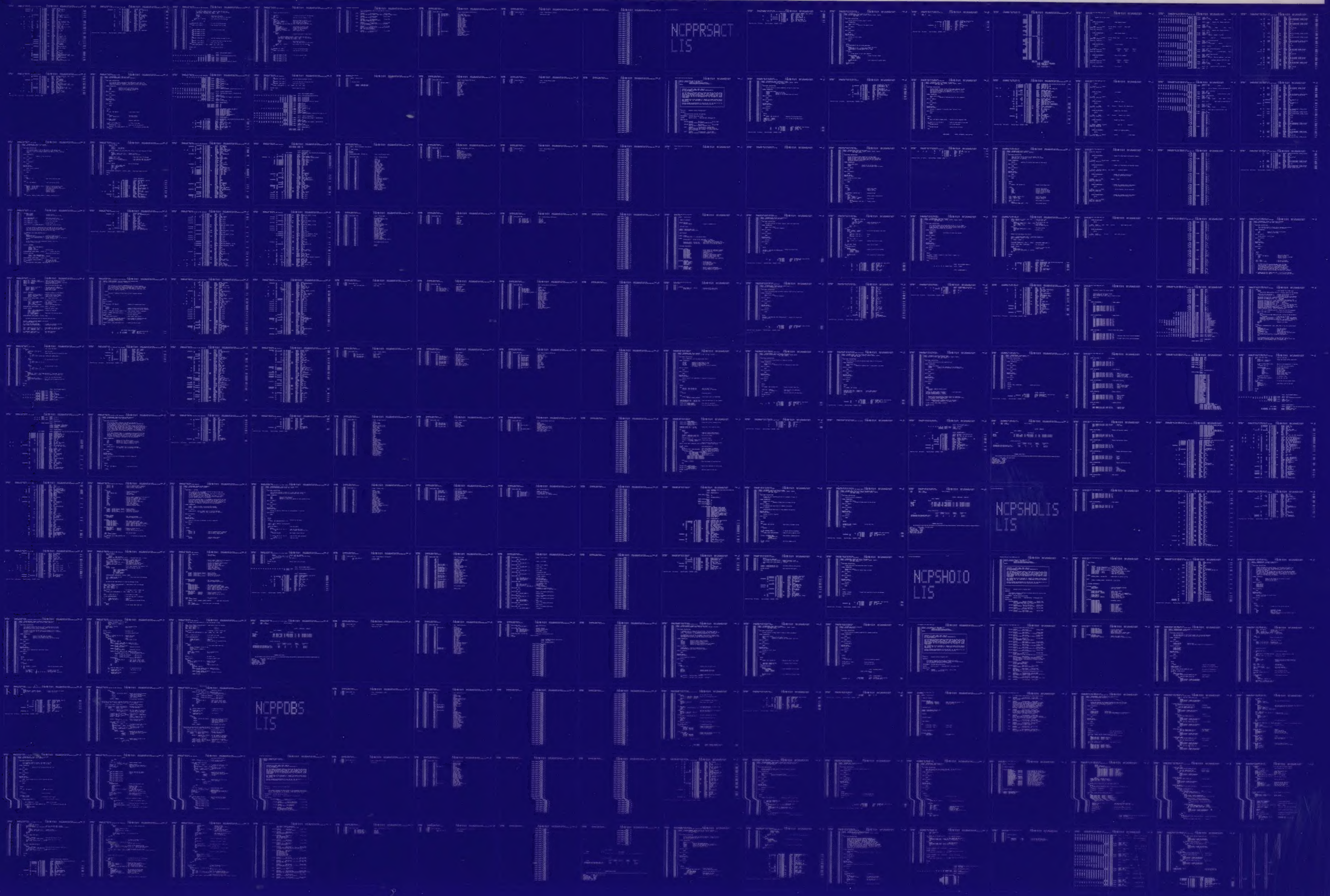
NMAHEAD
B32

NCPLIBRY
LIS

NMATAIL
B32

0268 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



NCPPRSACT
LIS

NCPSHOLIS
LIS

NCPSHOIO
LIS

NCPPDBS
LIS